AD-A153 039 HEATING PARAMETER ESTIMATION USING COAXIAL THERMOCOUPLE 1/2
GAGES IN WIND TUN..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. N T CAHOON
UNCLASSIFIED DEC 84 AFIT/GAE/AA/84D-3 F/G 9/2 NL

1.0 4.5 2.8 2.5 5.0 3.2 2.2 5.6 3.6 6.3 7.1 4.0 2.0 1.1 1.8 1.25 1.4 1.6

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

HEATING PARAMETER ESTIMATION USING COAXIAL THERMOCOUPLE GAGES IN WIND TUNNEL TEST ARTICLES

THESIS

Neil T. Cahoon
Captain, USAF

AFIT/GAE/AA/84D-3

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/AA/84D-3

HEATING PARAMETER ESTIMATION USING COAXIAL THERMOCOUPLE GAGES IN WIND TUNNEL TEST ARTICLES

THESIS

Neil T. Cahoon
Captain, USAF

AFIT/GAE/AA/84D-3

Approved for public release; distribution unlimited

AFIT/GAE/AA/84D-3

HEATING PARAMETER ESTIMATION USING COAXIAL THERMOCOUPLE GAUGES IN WIND TUNNEL TEST ARTICLES

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Aeronautical Engineering

Neil T. Cahoon, B.S.E.

Captain, USAF

December 1984

Acknowledgements

I wish to express my sincere thanks and gratitude to my thesis advisor, Major James K. Hodge for his expert guidance during the course of this study. His ability to provide engineering insight ("voodoo engineering") and open the doors of understanding never ceased. I would also like to say thanks for the encouragement when the obstacles appeared overwhelming.

Also to my wife, Shirley, for the sacrifices she made during the course of this academic experience, I would like to say thanks. She provided immeasurable support and encouragement during my long hours of study.

Table of Contents

List of Figures

List of Symbols

A, A', b, d	Coefficient Matrices (i x i)
c	Specific Heat
E	Residual Error Vector (m)
G	Kalman Gain Vector (i)
H	Thermocouple Location Matrix (m x i)
h_0	Magnitude of Heat Transfer Coefficient Ratio
h_s	Heat Transfer Coefficient Derivitive
h_{bar}	Heat Transfer Coefficient Ratio
h_{ref}	Reference Heat Transfer Coefficient at Zero State
I	Identity Matrix
J_k	Conditional Information Matrix
k	Thermal Conductivity
L	Total Number of Spatial Node Points
P	Covariance Matrix (i x i)
Q	Model Error Covariance Matrix (i x i)
q	Heating Rate
R_m	Covariance for mth measurement
S	Score Vector (k)
$S_{i,k}$	Sensitivity Vectors (i) for the kth Parameter
T_{aw}	Adiabatic Wall Temperature
t	Time
U	Temperature Vector (i)

x	Spatial Coordinate
Y	Thermocouple Measurement Vector
α	Angle of Attack
ε	Emissivity
θ	Parameter Vector
μ_n	Measurement Vector at nth Time Point
ρ	Density
σ	Stefan-Boltzmann Constant
ϕ	Transition Matrix
ϕ_c	Scaling Parameter for Specific Heat
ϕ_k	Scaling Parameter for Thermal Conductivity

<u>Superscripts</u>

$-$	a priori Propogation
$+$	a posteriori Propogation
$*$	Parameter Estimate
n	Time Level
s	Iteration Level
T	Transpose

<u>Subscripts</u>

i	Spatial Node Point
k	Number of Model Parameter
m	Number of Thermocouples
o	Freestream Conditions

Abstract

A heat energy balance is applied to a coaxial thermocouple gage for parameter estimation in wind tunnel test articles. This method can significantly reduce wind tunnel test costs and time. Modifications to the data reduction technique HEATEST (HEATing ESTimation) are made. The program allows for transient test techniques to be used as well as assuming an isothermal wall. A non-linear convective heat transfer coefficient model may also be used. Data is generated to test the new program. Temperature profiles throughout the thermocouple gage were good and were compared with changes in time step, thermocouple length, and number of discrete node points. The estimation of the convective heat transfer coefficient and thermal conductivity were excellent.

HEATING PARAMETER ESTIMATION USING COAXIAL THERMOCOUPLE GAUGES IN WIND TUNNEL TEST ARTICLES

I. INTRODUCTION

1.1 Background

The determination of heat transfer rates on hypersonic configurations upon reentry is important for the survival of the vehicle. The problem is that the heat rate is not a quantity which may be directly scaled from model tests in wind tunnels. However, the parameters which make up the heat rate equation (thermal conductivity and specific heat, for example) can be scaled from which the heat rate may then be calculated. Wind tunnel heat transfer measurements have traditionally used a thin walled test model fabricated with thermocouples mounted on the inside skin surface. The "thermal model" then yields a heat rate based on temperature measurement from the thermocouple. Another technique uses a coaxial thermocouple gage mounted in a thick skin model. A discussion of the two methods follows.

The Traditional Thin Skin Model

The "thermal model" of a traditional wind tunnel thin

skin model assumes that all of the heat penetrates the thin skin via conduction to a standard thermocouple gage mounted on the back face. No lateral conduction is assumed and since the emittance of the steel model is low, radiation is assumed negligible. A typical heat rate measurement data point is aquired by injecting the cooled model into the wind tunnel at a known temperature and time and by measuring the temperature at later times. The model is then removed from the tunnel, cooled, and a change in configuration is made in preparation for the next injection and subsequent dat point. There are three very severe limitations associate with this technique (Ref 1).The first is the inability to acquire more than one data point during any one injection. The thermal model simply does not allow for the type of change in configuration or model attitude which can be accomplished using dynamic testing techniques (to be discussed later). Associated with this limitation is the long cooling time between each test which significantly increases the cost per data point for the overall test. A second limitation is the special thin skin model which must be fabricated, further contributing to increased test cost. Finally, the assumption of no lateral conduction through the model may in fact be a poor assumption at some critical locations with large curvature. An alternate type of gage, the coaxial thermocouple, can eliminate these limitations with an overall effect of reducing time and cost.

A New Application For An Old Thermocouple

The coaxial thermocouple gage is shown in Fig. 1.1. It consists of a constantan (a metal alloy) jacket surrounding a chromel core with a thin layer of insulation separating the two metals. The coaxial gage is mounted in a steel model thick enough so that the thermal pulse is not sensed on the backface (ie. the model wall is considered a semi-infinite slab). The thermocouple surface is formed when the gage is lightly sanded to match the contour of the model. Some gages are available with backface temperature monitoring to assure that the thermal pulse does not reach the backface in any given run so that an analytical integration of the heat equation can be used to determine the heating rate history. The backface temperature information prior to this investigation is not factored into the data reduction process, however.

Operation of the coax gage is based on uniform conduction along the gage length which would necessitate the model be made of a material with similar thermal properties (Ref 3). The thermal properties of stainless steel match very closely with the gage properties, therefore, the presence of the gage is negligible. The matching of thermal properties also enhances accuracy. A coaxial gage which is matched thermally with the model allows an isothermal wall assumption, whereas other gauges such as calorimeters and thin film gages are not thermally matched, and cause a non-isothermal wall. Measured heat transfer can be in error by

$$\{\dot{U}\} = [A]\{U\} + \{b\} + W(t)$$

$$\{\dot{S}_k\} = [A]\{S_k\} + \{d_k\} \tag{3-1}$$

These equations are solved using a tridiagonal algorithm in subroutine TPS3 for the temperature states, and subroutine SENS for the sensitivity of the temperature to the kth parameter. Propogation of the covariance, P, of the temperature state at each node is accomplished by the approximate difference equation,

$$P(t_n^-) = \phi(\Delta t)P(t_{n-1}^+)\phi^T(\Delta t) + \int_{t_{n-1}}^{t_n} \phi(t_n-\lambda)Q\phi^T(t_n-\lambda)\,d\lambda \tag{3-2}$$

where ϕ is the transition matrix and where the - and + superscripts are used to denote the expected values before an update (or a priori) and updated (or a posteriori) values, respectively. This calculation is made in subroutine TPSOSP2.

A model of the temperature measurement process must be used for the Kalman filter equations. The measurement equation to identify thermocouple location is,

$$Y(t_n) = H\,\{U(t_n)\} + \{\mu_n\} \tag{3-3}$$

The updated temperature is calculated by,

$$U(t_n^+) = U(t_n^-) + GE(t_n) \tag{3-4}$$

time. They are found by employing a Kalman filter - a set of recursive equations that optimally combine the propogation of the model equations with measurement updates at each sample time. After the entire temperature - time (state) history has been calculated, a gradient algorithm is used to solve for best estimates of the parameters according to a maximum likelihood criterion.

The second type of estimate consists of the parameters defined in the parameter vector,$\{\Theta\}$, as given in Equation 2-10. These parameters remain essentially constant throughout a transient maneuver profile such as a pitch sweep and are estimated based on data from the entire maneuver history. The process of estimating states and parameters is then iterated for convergence to some optimal estimate.

The method used to estimate the states and parameters is formulated from stochastic estimation theory and is known as adaptive estimation. A detailed development of the estimation equations is beyond the scope of this thesis and the reader is referred to References 7 and 8 for more detail.

The thermal model equations for temperature and sensitivity have already been written in the matrix stochastic estimation form of Equation 2-12 as,

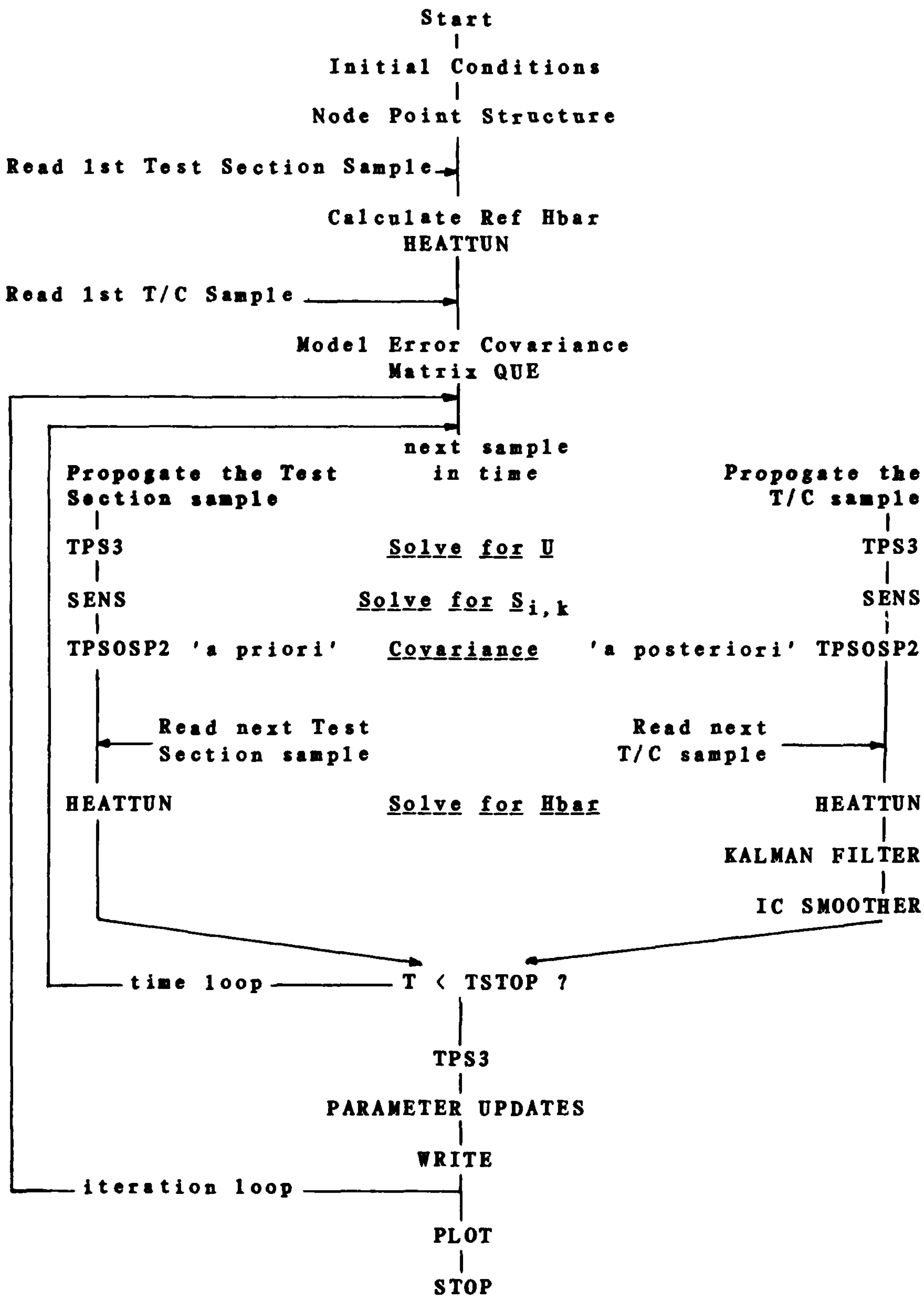


Figure 3.1 HEATEST Algorithm Summary

III. HEATEST OVERVIEW

The HEATEST program was originally developed to determine and model heat rates from the Space Shuttle Orbiter thermocouple data, hence most of its nomenclature references flight data samples and trajectory samples. The wind tunnel equivalence of the trajectory sample would be the test section conditions at the time of the sample (ie. density, velocity, pressure, etc.). The flight data are the thermocouple measurements from the coaxial gages.

An algorithm summary of the HEATEST program is given in Figure 3.1. The initial conditions for the temperature profile, U(t), and the initial covariance, P(t), are specified at the start of the wind tunnel test run. Heating model initial parameters, and the initial reference values for the heating model are read in as inputs to the program. Initial sensitivities of the state are specified to be zero. The node point structure throughout the depth of the thermocouple is then calculated from the input of the length of the gage and the number of node points.

Two types of estimates must be made in order to describe the thermodynamic environment in the wind tunnel. The first type are the state estimates, which are defined by each node temperature. These state estimates are not constant since the temperature varies throughout the maneuver, and hence, must be estimated at each node point in

The {b} vector is not used directly for the covariance equation, but is approximated by an error model given by,

$$Q_{error} = [h_{bar} h_{ref} (T_{aw} - U_1) \Delta x / \phi_k k]^2 \qquad (2\text{-}15)$$

where S is the sensitivity. The i subscript identifies the node point and the second subscript identifies the particular parameter number. The sensitivity equations may also be written in the familiar form,

$$[A']\{S_{i,k}^{n}\} + \{d\} = 0 \tag{2-12}$$

The sensitivity equations are developed and summarized in Appendix B.

2.3 Covariance Equation

Propogation of the covariance of the temperature state at each node requires the equations to be of the form

$$\begin{aligned} \{\dot{U}\} &= [A]\{U\} + \{b\} \\ \text{and} \quad \{\dot{S}_{i,k}\} &= [A]\{S_k\} + \{d_k\} \end{aligned} \tag{2-13}$$

Substituting the definitions of Equations (A-14) into Equations (2-7) and (2-8) and rearranging yields a common tridiagonal [A] matrix for the above equations which is shown presently,

$$[A] = \begin{bmatrix} -\dfrac{[RM_1+RP_1+4\varepsilon\sigma(U_1^{n,s})^3+h_{bar}h_{ref}]}{RCX_1} & \dfrac{RP_1}{RCX_1} & 0 \\ \ddots & \ddots & \ddots \\ \dfrac{RM_i}{RCX_i} & -\dfrac{(RM_i+RP_i)}{RCX_i} & \dfrac{RP_i}{RCX_i} \\ & \ddots & \ddots \\ 0 & & \end{bmatrix}$$

(2-14)

$$[A']\{U_i^n\} + \{b\} = 0 \tag{2-9}$$

where $[A']$ is an n x n tridiagonal matrix of material properties and $\{U_i^n\}$ is the n-dimensional column vector of unknown temperature at each node point for each time.

In general, the unknown parameters in this model formulation are the heat transfer coefficient intercept, h_0, the slopes $h_{\alpha 1}$ and $h_{\alpha 2}$, and the scaling parameters for specific heat and thermal conductivity, ϕ_c and ϕ_k, respectively. These parameters may be defined as a vector, , of unknown parameters for use in the system identification scheme as,

$$\vartheta = \{h_0, h_{\alpha 1}, h_{\alpha 2}, \phi_c, \phi_k\}^T \tag{2-10}$$

The primary purpose of the heating estimation program is to obtain best estimates of these parameters during transient test maneuvers. To estimate these parameters it is necessary to calculate the model sensitivity to each unknown parameter.

2.2 Sensitivity Equations

The derivative of Equation (2-7) with respect to each parameter yields equations of the same form as Equation (2-9) from which the HEATEST program propogates the sensitivity. For example, the sensitivity of the temperature with respect to h_0 would be written as follows,

$$\frac{\partial U}{\partial \theta_1} = \frac{\partial U}{\partial h_0} = S_{h0} = S_{i,1} \tag{2-11}$$

level defined by the superscript n,

$$(U_1^{n,s+1})^4 = (U_1^{n,s})^4 + 4(U_1^{n,s})^3[U_1^{n,s+1} - U_1^{n,s}]$$

$$= -3(U_1^{n,s})^4 + 4(U_1^{n,s})^3 U_1^{n,s+1} \qquad (2\text{-}6)$$

Substituting Equation (2-6) into (2-5) yields,

$$\frac{\rho\phi_c c\Delta x}{2}\,\frac{U_1^n - U_1^{n-1}}{\Delta t} = \frac{-\phi_k k_{1+1/2}}{\Delta x}U_1^n + \frac{\phi_k k_{1+1/2}}{\Delta x}U_2^n$$
$$- \varepsilon\sigma[-3(U_1^{n,s})^4 + 4(U_1^{n,s})^3 U_1^{n,s+1} - (U_\infty^n)^4]$$
$$+[h_0 + h_{\alpha 1}(\alpha-\alpha_1) + h_{\alpha 2}(\alpha-\alpha_2)]h_{ref}(T_{aw} - U_1^{n,s+1})$$

$$(2\text{-}7)$$

The model equation for the interior node points, (i=2,imax), yields,

$$\rho\phi 1_c c\Delta x\,\frac{U_i^n - U_i^{n-1}}{\Delta t} = \frac{\phi_k k_{i-1/2}}{\Delta x}U_{i-1}$$
$$- \frac{\phi_k(k_{i-1/2} + k_{i+1/2})}{\Delta x}U_i$$
$$+ \frac{\phi_k k_{i+1/2}}{\Delta x}U_{i+1}$$

$$(2\text{-}8)$$

Equations (2-7) and (2-8) can be rearranged into the familiar matrix form,

parameters other than those included in the reference heat transfer coefficient are summarized by the static transfer relation or heat transfer coefficient ratio, $h_{bar} = h/h_{ref}$. Here, the ratio is assumed to be piecewise linear with respect to angle of attack as derived from Lagrange Interpolation Theory (Ref 6 and Ref therin).

$$h_{bar} = [h_0 + h_{\alpha 1}(\alpha - \alpha_1) + h_{\alpha 2}(\alpha - \alpha_2)] \qquad (2\text{-}4)$$

where h_0 is the magnitude of the heat transfer coefficient, h at the reference conditions ,α_1 , specified by the one subscript. The heating parameters h_0 , $h_{\alpha 1}$, and $h_{\alpha 2}$ are considered to be unknown and constant over a prescribed time period, and will be estimated by the HEATEST program. The parameters correspond to a derivative with respect to deflection angle of the model. Thus, for constant step size, the model equation at the surface node, (i=1), becomes,

$$\frac{\rho \phi_c c \Delta x}{2} \frac{U_1{}^n - U_1{}^{n-1}}{\Delta t} = \frac{-\phi_k k_{1+1/2}}{\Delta x} U_1{}^n$$

$$+ \frac{\phi_k k_{1+1/2}}{\Delta x} U_2{}^n - \varepsilon\sigma[(U_1{}^n)^4 - (U_\infty{}^n)^4]$$

$$+ [h_0 + h_{\alpha 1}(\alpha - \alpha_1) + h_{\alpha 2}(\alpha - \alpha_2)] h_{ref}(T_{aw} - U_1{}^n) \qquad (2\text{-}5)$$

The non-linear radiation term is quasi-linearized on an iteration level defined by the superscript s and by the time

where ε radiative emissivity
σ StefanBoltzmann constant
c material Specific Heat
ρ material density
k Thermal Conductivity
ϕ_c Specific Heat scaling parameter
ϕ_k Thermal Conductivity scaling parameter

The material specific heat and thermal conductivity are both scaled by the two factors ϕ_c and ϕ_k, respectively, hence the value for c and k will remain unchanged. The parameters ϕ_c and ϕ_k will be estimated by the HEATEST program. Coefficients with subscripts which are less than one or greater than n are zero. The radiation and heat rate terms are also zero except at the surface node. Equation 2-1 includes terms due to conduction from adjacent node points $k_{i-1/2}/\Delta x_{i-1}$, surface radiation $\varepsilon\sigma U_i^4$, and the convective transfer of energy as obtained from the heating model. The resulting system of implicit difference equations must be solved simultaneously.

The heating model for the convective transfer of energy is based on Newton's Law of Cooling,

$$q = h(T_{aw} - T) \tag{2-2}$$

Non-dimensionalizing by a reference heat transfer coefficient, h_{ref} yields,

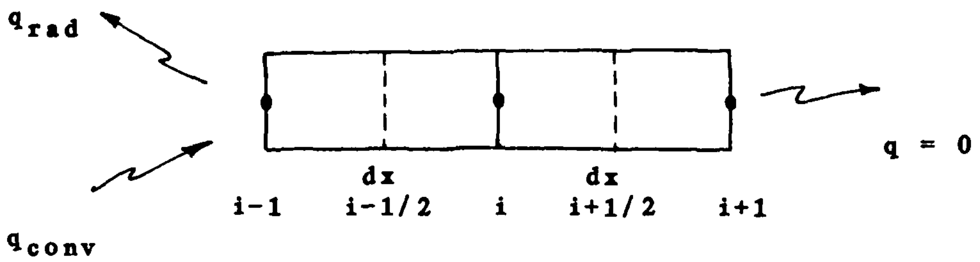
$$q = h_{bar} h_{ref} (T_{aw} - U_i) \tag{2-3}$$

The dependance of the heat transfer coefficient on

II. THE THERMAL MODEL

2.1 Temperature Equations

A cross section of the one-dimensional model is given in Figure 1.1 and below as a typical coaxial thermocouple gage.



An energy balance is performed on each element. The thermal conductivity, k, is taken as an average between each node. Fourier's Law of Heat Conduction throughout the gage, the Stefan-Boltzmann Law for radiation and Newton's Law of Cooling for convection on the surface face yield a system of n nonlinear differential equations of the form:

$$\begin{aligned}
&[(\rho_i \phi_c c_i \Delta x_i + \rho_{i-1} \phi_c c_{i-1} \Delta x_{i-1})/2][(U_i^{n} - U_i^{n-1})\Delta t] \\
&\quad = \phi_k [k_{i-1/2}/\Delta x_{i-1}] U_{i-1}^{n} \\
&\qquad - \phi_k [k_{i-1/2}/\Delta x_{i-1} + k_{i+1/2}/\Delta x_{i+1}] U_i^{n} \\
&\qquad + \phi_k [k_{i+1/2}/\Delta x_{i+1}] U_{i+1}^{n} \\
&\qquad - \varepsilon\sigma(U_i^{4} - U_{i-1}^{4}) + h_{bar} h_{ref} (T_{aw} - U_i) \qquad (2\text{-}1)
\end{aligned}$$

gage using a Kalman filter(Ref 2). The purpose of this investigation is to incorporate the appropriate thermal model equations into the HEATEST program for application to coaxial thermocouple gages as used in the wind tunnel. Two reasons for doing this are, 1) replace the analytical with the semi-infinite assumption, ie. extend the run time or shorten the gage, and, 2) estimate thermal properties. Testing and verification of the modified program is necessary for verification of the validity of the results. Simulated data are generated by an analytical solution, and are processed for testing purposes for which the results are known.

1.3 Overview

A development of the pertinent temperature, sensitivity, and covariance equations will be developed for introduction into the HEATEST program in Chapter II. Details in format for programming may be found in Appendix A & B. Chapter III is an overview of the HEATEST algorithm and shows how the equations developed in Chapter II are utilized. Chapter IV outlines the method for testing the program and offers a discussion of the test cases made and results. Finally, conclusions about the validity of the modifications, and suggestions for further improvement are included in Chapter V.

up to 40% because of the non-isothermal (Ref 2). The same rugged model built for pressure measurements can be used for temperature measurements as well, which would further reduce the wind tunnel costs. The data reduction technique, which uses the temperature time history, eliminates the requirement to fix the model configuration or attitude during any one run, hence dynamic testing techniques may be used similar to the flight test technique used for the Space Shuttle Orbiter (Ref 4). The model may be swept in angle of attack, for example, to determine heat rates as a function of angle of attack. All of these attributes along with a short response time and no required calibration (ref Knox) combine to yield the wind tunnel engineer a tool of marked improvement over previous methods.

1.2 OBJECTIVES

A method of analysis to identify the aerothermodynamic flight environment and update the thermal model of the engineering simulation of the Space Shuttle Orbiter was designed by the Air Force Flight Test Center. This method is in the form of a digital computer program called HEATEST (HEATing ESTimation). The program provides a correlation of the heating as well as a heat rate time history. The program integrates numerically instead of relying on some analytical assumption. It satisfies a maximum likelihood criteria for each parameter and obtains best estimates for the temperature at discrete nodes throughout the length of the

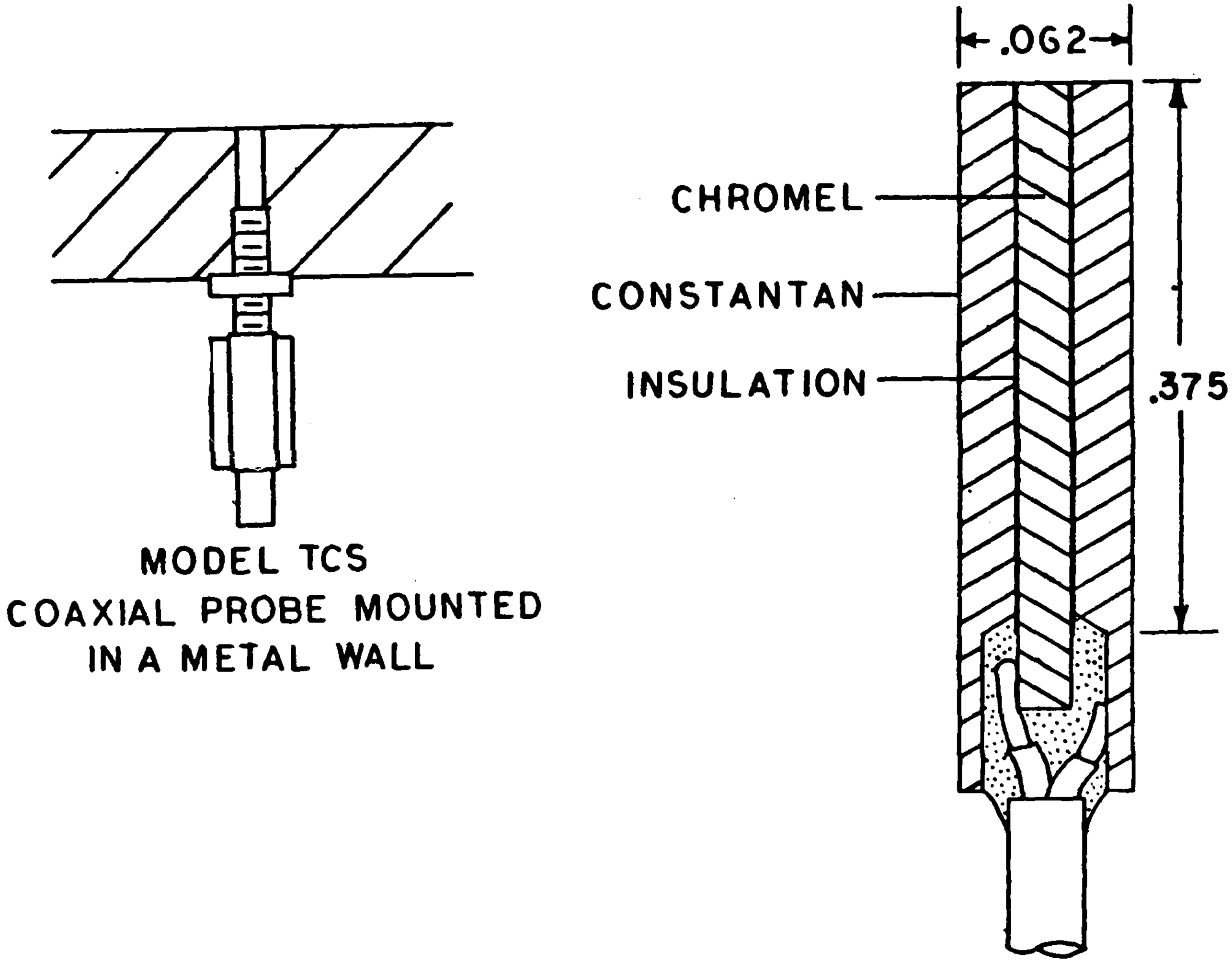


FIGURE 1.1 COAXIAL THERMOCOUPLE GAGE

where

$$G = P(t_n^-)H^T[HP(t_n^{-1})H^T + R_m]^{-1}$$

$$E = Y(t_n) - HU(t_n^-)$$

The updated sensitivities are calculated by,

$$S_k(t_n^+) = [I-GH]P(t_n^-)[I-GH]^T + GR_mG^T \qquad (3\text{-}5)$$

The updated covariance is calculated by,

$$P(t_n^+) = [I-GH]P(t_n^-)[I-GH]^T + GR_mG^T \qquad (3\text{-}6)$$

To alleviate the problem of imprecise initial conditions, a fixed point smoothing algorithm has been added to the HEATEST program. Details of the smoother and its effects in the adaptive estimation scheme may be found in Reference 6.

Finally, the best estimates of the parameters are then estimated at the end of a specified time segment by the gradient algorithm,

$$\theta^* = \theta - [\partial^2F/\partial\theta^2]^{-1}\partial F/\partial\theta = \theta + J^{-1}S \qquad (3\text{-}7)$$

where,

$$J_{i,j} = \sum_{n=1}^{N} S_{i,k}(t_n^-)H^T[HP(t_n^-)H^T+Rm]HS_{i,k}(t_n^-)$$

$$S_k = \sum_{n=1}^{N} S_{i,k}(t_n^-)H^T+Rm]^{-1}[Y(t_n)-HU(t_n^-)]$$

The matrix J is an approximation for the Jacobian or conditional information matrix and is given in component

form by $J_{i,k}$. The score vector, S_k, is used to approximate the gradient of the likelihood function for a large number of time samples.

Using these equations, best estimates for the temperature time history (states) at each node can be found, as well as the deviation in temperature as provided by the covariance matrix. Also, an estimate of the parameter uncertainty is provided by the Cramer-Rao bound. The Cramer-Rao bound relates the conditional information matrix to the covariance of the parameter estimate.

IV. RESULTS

4.1 Test Procedure

To test the validity of the program modifications, a set of contrived data was generated. It's development assumes that the heat rate due to convection at the surface node is constant and equal to the heat rate due to conduction at the surface. The heat rate due to convection is given by,

$$q = h(T_{aw} - T_w)$$

or

$$q = h_{bar} h_{ref} (T_{aw} - T_w) \qquad (4-1)$$

where h_{bar} is defined as in Equation A-12. The equation for the heat rate due to conduction assuming a one-dimensional, homogenous, semi-infinite solid is as follows (Ref 9),

$$q = \frac{(\rho c k)^{1/2}}{\pi} \int_o^t \frac{dT_w(\tau)}{d\tau} \frac{d\tau}{(t-\tau)^{1/2}} \qquad (4-2)$$

where t = time from start of heating
T(t) = surface temperature rise
τ = dummy variable of integration

Equating Equations 4-1 and 4-2 yields,

$$q = h_{bar}h_{ref}(T_{aw}-T_w) = \frac{(\rho ck)^{1/2}}{\mu} \int \frac{dT_w(\tau)}{d\tau} \frac{d\tau}{(t-\tau)^{1/2}} \tag{4-3}$$

Two different expressions for the derivative of the wall temperature with respect to time were used. The first implied a linear change in temperature with respect to time yielding a constant for dT_w/dt and the second expression assumes that temperature was a quadratic function of time as shown,

<u>Linear</u>	<u>Quadratic</u>
$T_w = bt + c$	$T_w = at^2 + bt + c$
$\frac{dT_w}{dt} = b$	$\frac{dT_w}{dt} = 2at + b$

Solving Equation 4-3 for T_{aw} so that h and q are constant and after making the indicated substitutions and integrating yields,

<u>Linear assumption</u>

$$T_{aw} = T_w + \frac{2b}{h_{bar}h_{ref}}\sqrt{\frac{\rho ckt}{\pi}} \tag{4-4}$$

<u>Quadratic assumption</u>

$$T_{aw} = T_w + \frac{2}{h_{bar}h_{ref}}\sqrt{\frac{\rho ckt}{\pi}}\left(b + \frac{4at}{3}\right) \tag{4-5}$$

A short computer program was written to produce a temperature-time history in the data tape format for the HEATEST program. For the above equations, h_{ref} and h_{bar} (the estimated parameter) were set equal to 1 and the coefficients a, b, and c, were selected to yield reasonable values for T_w.

4.2 Test Cases

The reference test case was taken to be a 10 sec. simulated wind tunnel test run using the linear data provided from the previous section. Thermocouple samples and test section samples were provided at the rate of one sample per second. The objective was to examine the rate of convergence of the temperature states and to estimate the first parameter, h_0. Recall from Section 4.1 that the data was generated to yield a value of one for h_0. Also of interest, was the validity of the model to the semi-infinite solid assumption (ie. no change in the temperature at the back face node throughout any specified time segment).

The input data is shown in Figure 4.1 and is generated digitally depending upon a desired time step (del t). The initial temperature throughout each gage is assigned a value of 60^{o}F. Figure 4.2 shows the input temperature values for a del t = 1 sec. as used in the reference test case.

Figure 4.3 identifies the temperature state at the 2 sec.(lowest curve), 6 sec.(middle curve), and 10 sec.(top curve) times following the first iteration through the

updated HEATEST program. It clearly shows that at the back face node, the semi-infinite solid assumption used to derive the data is violated. This is indicated by the change in backface (node 6) temperature with time. Note also, however, that the temperature gradient at the back nodes (between nodes 5 and 6) is zero due to the adiabatic wall assumption. It should also be pointed out that surface node temperature response to the given input was immediate with no time lag.

Figure 4.4 is similar to Figure 4.3 except the temperature states and parameter estimates have been iterated to convergence, in this case, three times. Overlaying the two figures shows no perceptible difference between them, and the data shows no variations in values until after the second decimal point. In spite of the response of the back face node which would ordinarily invalidate the test, the estimated value for h_0 was .99598, within .4% of the desired value of 1! Several test cases will be compared to this reference by examining changes in time step, thermocouple length, and the number of node points. Also, an examination of the ability of the program to estimate the other parameters, follows.

4.2.1 Changes in Time Step

Figures 4.4, 4.5, and 4.6 show temperature states at 2, 6, and 10 sec. for del t equal to 1, .5, and .25 sec., respectively. All three curves required three iterations

for convergence. The curves are all similar in shape with almost no perceptible differences. However, if the 10 sec. curves from del t = 1 sec. and del t = .25 sec. are overlayed as in Figure 4.7, a small difference may be noted at the backface nodes indicating that, indeed, a decrease in time step will yield a profile which will more closely approximate the model.

4.2.2 Changes in Thermocouple Length

Changes in thermocouple length offered the most dramatic changes in temperature state as can be seen in Figures 4.8, 4.9, and 4.10 where the lengths range from .1 ft., .05ft., and .025 ft., respectively. The two longer lengths converged within three iterations while the short thermocouple length took four iterations to converge. The extra iteration is most likely due to the large deviation from the model and the large differences in temperature state from one time step to the next. Figure 4.11 compares the temperature state of each thermocouple length after the 10 sec. run. The lowest curve is associated with the longest thermocouple, and the upper curve is associated with the short gage.

4.2.3 Changes in Number of Node Points

For this comparison, a 10 sec. run with 6 node points and a 3 sec. run with 6 node points (Figures 4.12 and 4.13) will be compared with a 10 sec. run with 12 node points and

a 3 sec. run with 12 node points (Figures 4.14 and 4.15). Each of the test runs were converged by the third iteration. At both of the different run times, increasing the number of nodes yielded a solution which more closely approximates the model (ie. the semi-infinite solid at the back face node) and gave correspondingly better estimates for h_0. The comparison of temperature states is better represented by Figures 4.16 and 4.17 which directly compares 6 nodes and 12 nodes interspersed evenly throughout the .05 ft. long thermocouple for 10 sec. and 3 sec. run times, respectively.

4.2.4 Parameter Estimation

The ability of the algorithm to estimate h_0 has already been discussed. To summarize, even when the output temperature states clearly violate the semi-infinite solid assumption used in generating the data, the estimated values of h_0 remain within 15%. The 15% is a worst case number derived from the short thermocouple using a course grid for a long run time. An average deviation which considers all of the test cases evaluated is closer to 3%.

To estimate ϕ_k, an erroneous data value was given for K_{data}, wherin the program iterated to a value for ϕ_k which, when multiplied by the erroneous K_{data}, would yield the correct value, ie. $\phi_k K_{data} = K_{correct}$. The erroneous K_{data} which was input into the program was 14% in error of the $K_{correct}$ value. The value provided by the program for ϕ_k when multiplied by K_{data} yielded a value within .9% of

the $K_{correct}$ value!

The estimate of ϕ_c was not as successful, however. After 6 iterations, the solution was diverging from the expected value. A suspected sign error in the ϕ_c sensitivity calculation is the most likely cause.

The quadratic input data was used as a comparison to the linear data to challenge the algorithm, ie. the more complicated the input the more difficult the estimation process. No direct comparison may be made of temperature, however, since the input data is different. The parameter estimation of h_ϕ using the quadratic input data was still excellent yielding .1%, while the estimate using linear data was somewhat better at .05%.

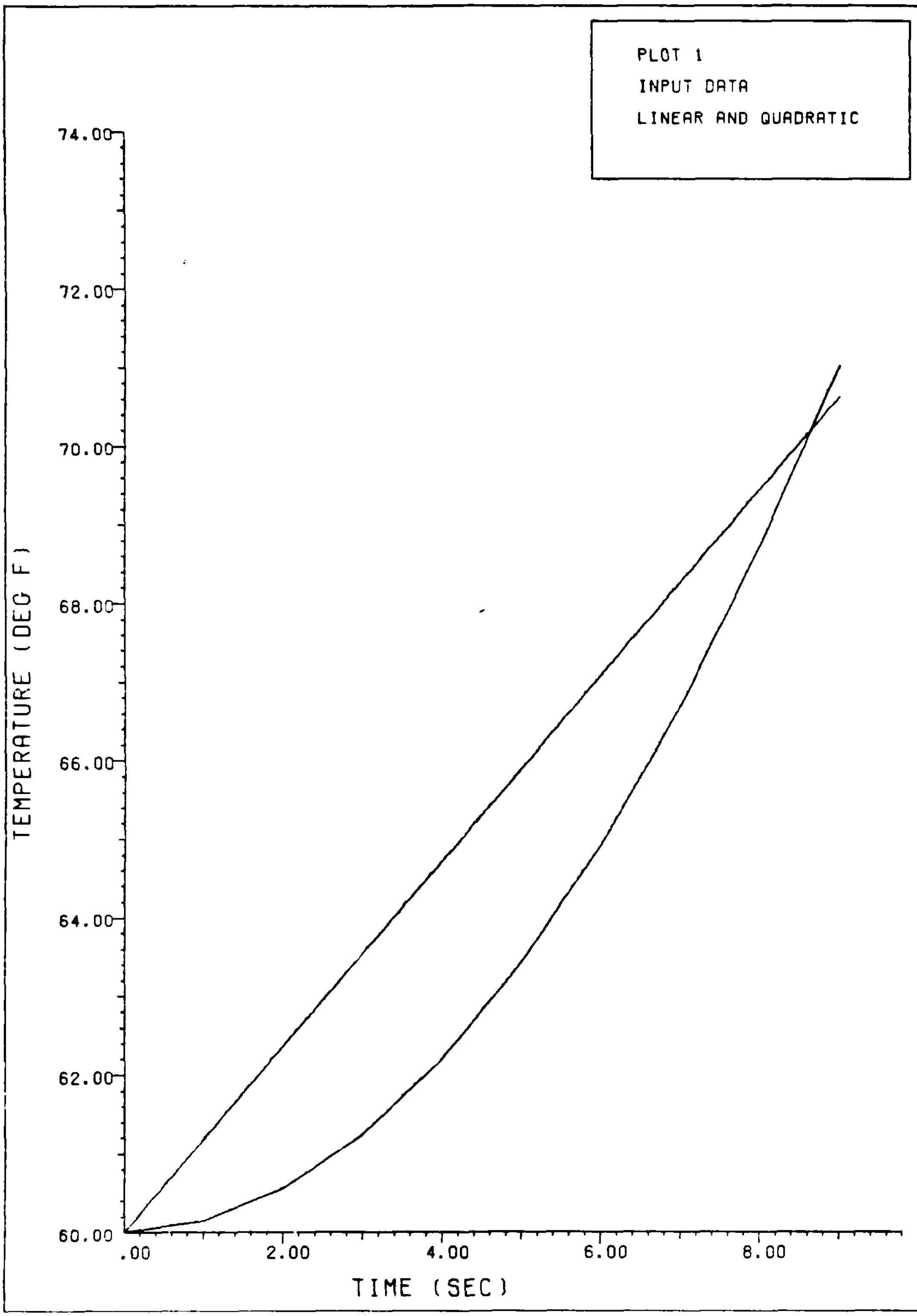

PLOT 1
INPUT DATA
LINEAR AND QUADRATIC
74.00
72.00
70.00
68.00
66.00
64.00
62.00
60.00
TEMPERATURE (DEG F)
.00
2.00
4.00
6.00
8.00
TIME (SEC)

FIGURE 4.1

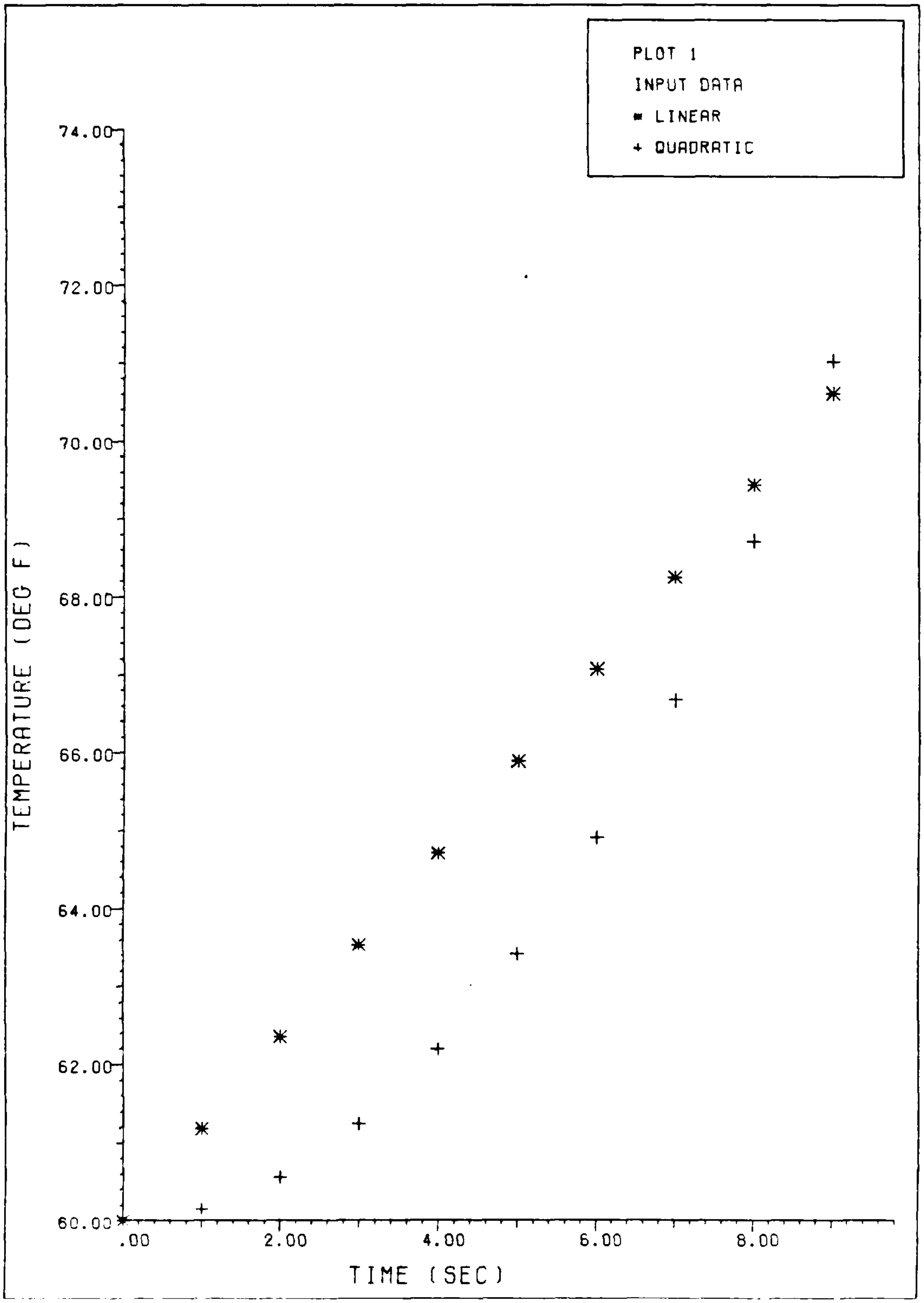

PLOT 1
INPUT DATA
* LINEAR
+ QUADRATIC
74.00
72.00
70.00
68.00
66.00
64.00
62.00
60.00
TEMPERATURE (DEG F)
.00
2.00
4.00
6.00
8.00
TIME (SEC)

FIGURE 4.2

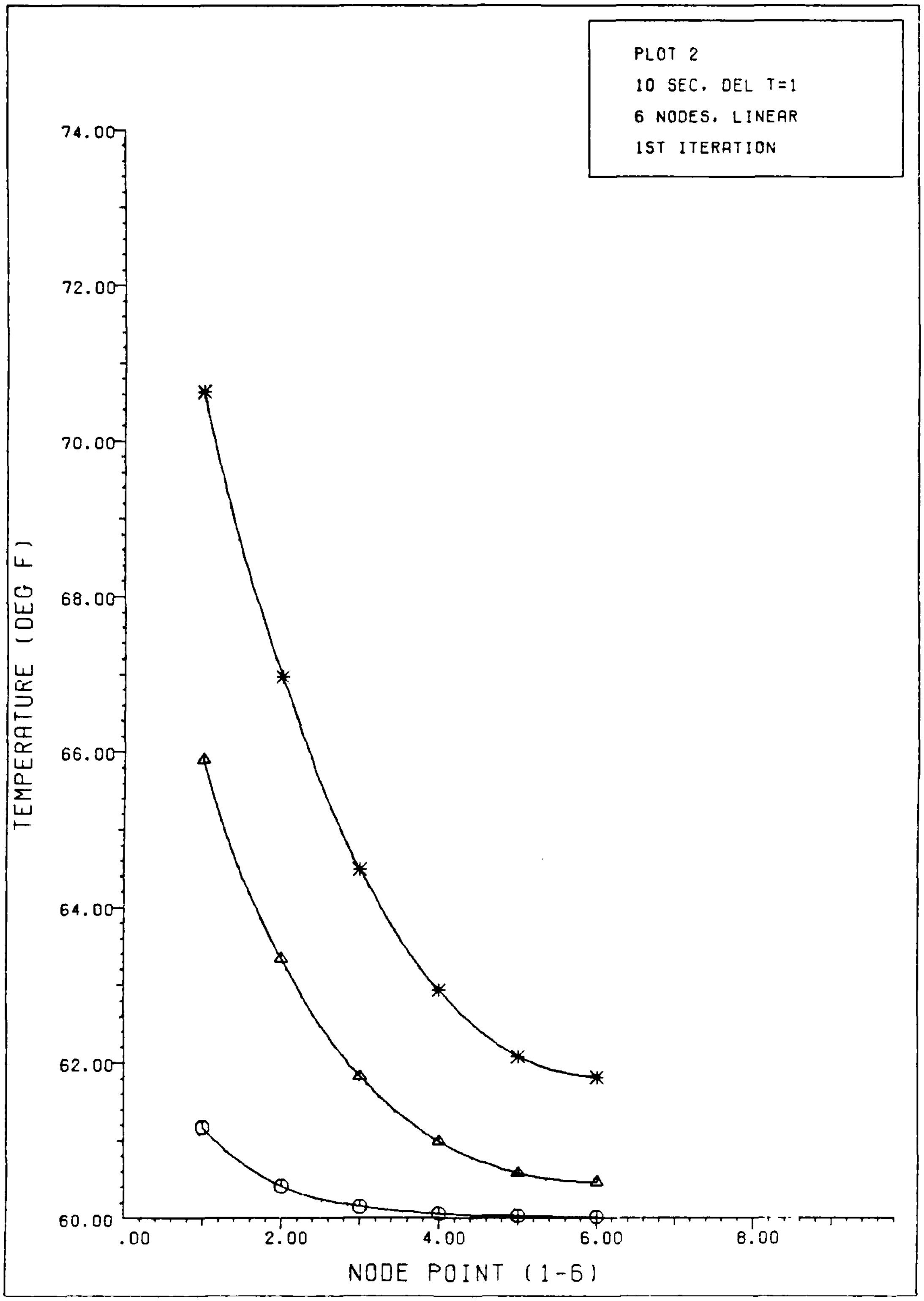


FIGURE 4.3 TEMP VS NODE POINT HISTORY (2,6,10 SEC)

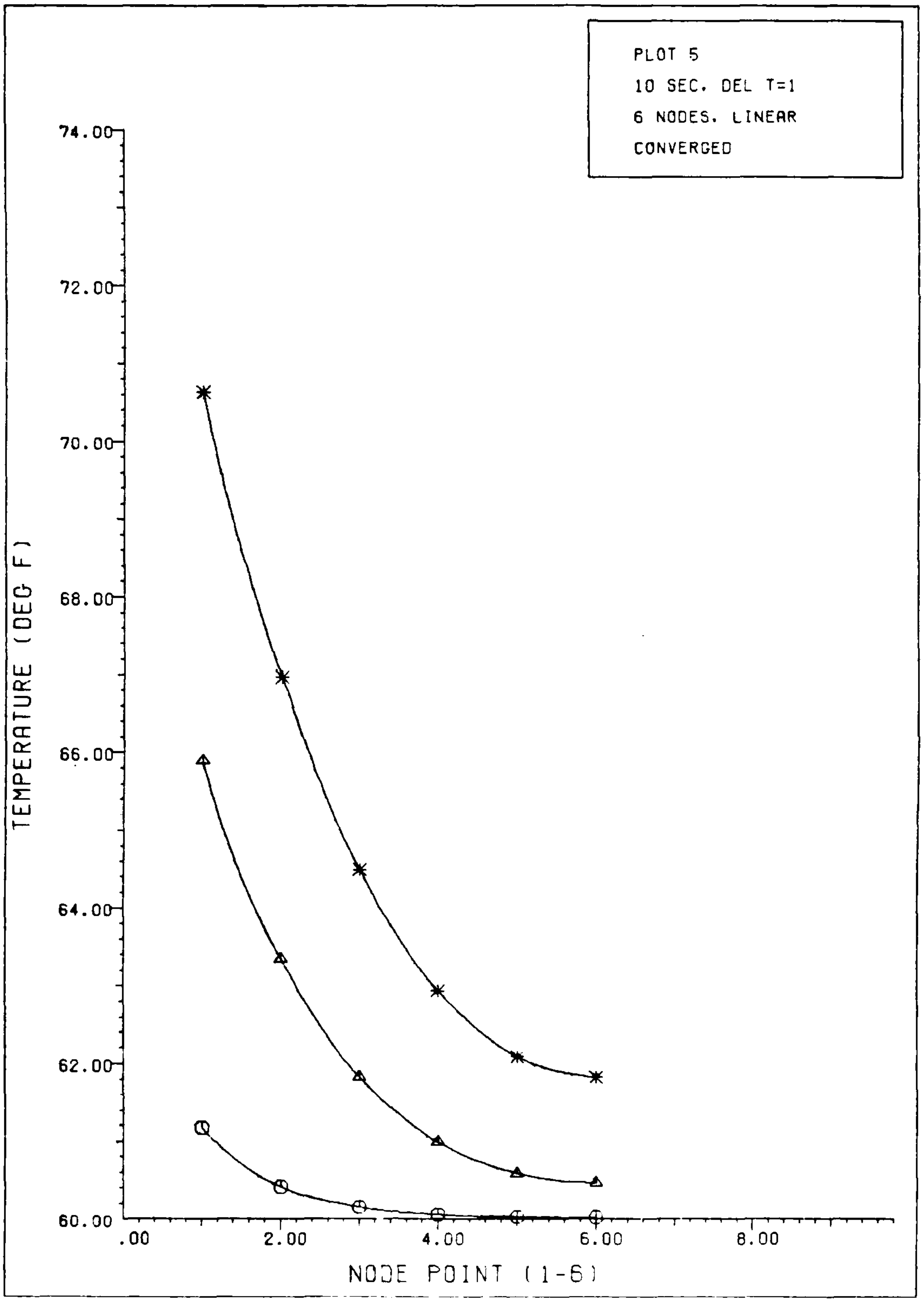


FIGURE 4.4 TEMP VS NODE POINT HISTORY (2.5.10 SEC)

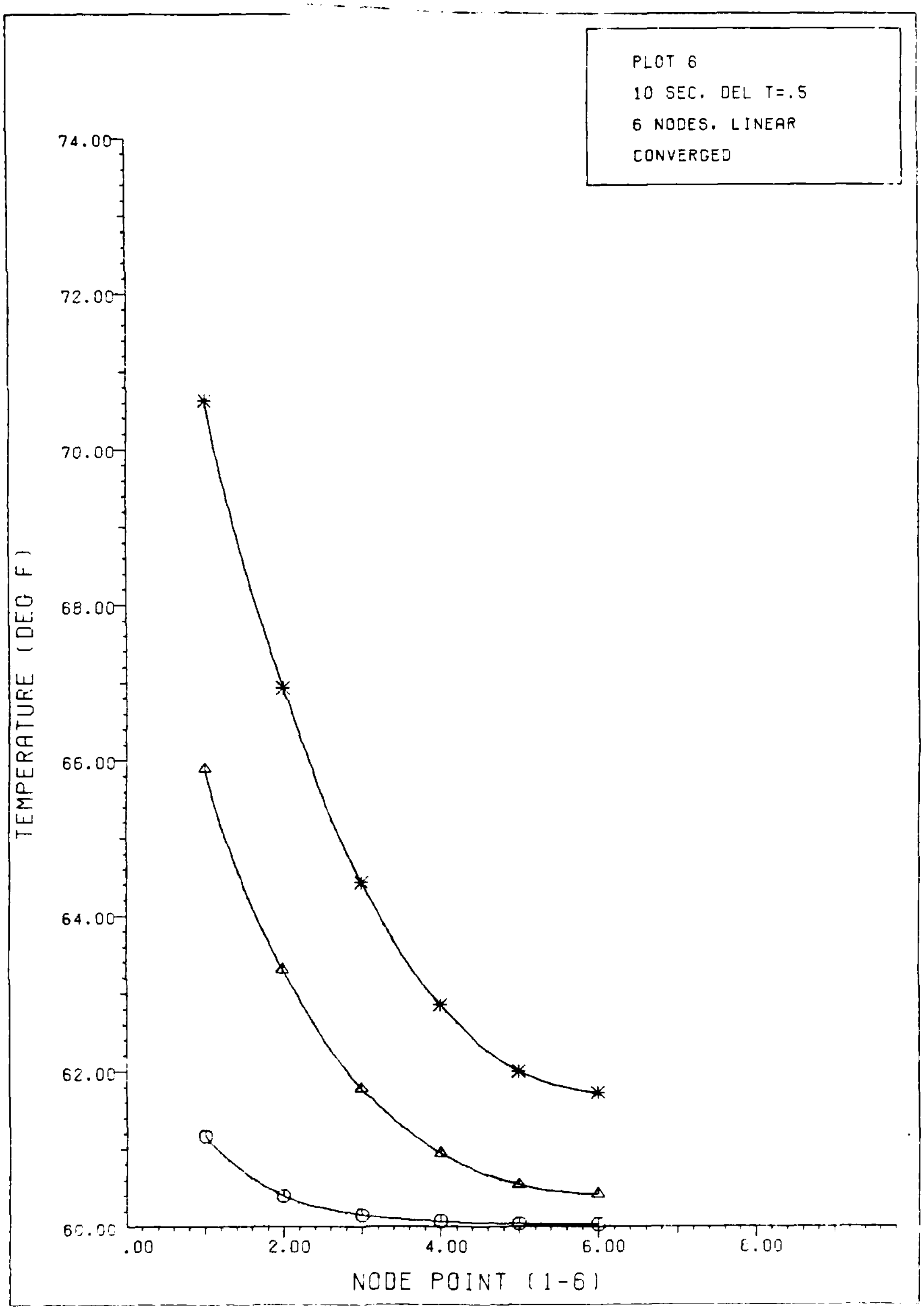


FIGURE 4.5 TEMP VS NODE POINT HISTORY (2,6,10 SEC)

the utility of the program to be used in wind tunnel runs of much longer duration.

5.2 Recommendations

Actual wind tunnel test data from coaxial gages needs to be analyzed by the program to instill more confidence in the results. This would require that the data tape format be modified to be compatible with the program inputs.

Also, a prescribed model for h_{bar} as a function of angle of attack needs to be input to determine the ability of the program to estimate the piecewise linear derivitives, $h_{\alpha 1}$ and $h_{\alpha 2}$. The additional input data would then be the wind tunnel model angle of attack at each thermocouple sample time.

Another potential modification to the program would be to use a second order time derivitive approximation as opposed to the current first order approximation. It is suspected that the increased accuracy would improve the state estimates particularly for large time gaps in thermocouple data.

The temperature state estimates might be improved by incorporating a variable grid. An exponential grid generation scheme would concentrate node points near the surface where the largest temperature gradients exist.

A final recommendation, would be to make the program more 'user friendly' and to publish documentation much like a users manual.

V. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The program works for this simplified test case of known parameters and number of node points. The ability of the program to determine the temperature state appeared very good but definite conclusions cannot be drawn without comparing this data to actual thermcouple data. The ability of the program to estimate the parameters, h_0 and ϕ_k was demonstrated extremely well. Re-examination of the ϕ_c equations must be made before further conclusions may be made about the ability of the program to estimate this parameter. It _is_ theoretically possible, however.

The objective, to validate the use of the modified HEATEST program for use of coaxial thermocouple gages on wind tunnel test articles, has been met for the special case of this analytical model. The program is capable of determining temperature states and estimating parameters with a high degree of accuracy. A note concerning the semi-infinite slab assumption needs to be emphasized. This assumption was made only for the data generation program to yield data for which the analytical solution was known. As mentioned in Chapter 1, some coaxial gages are available with backface temperature monitoring which would also provide another temperature measurement to enhance estimation of ϕ_k and ϕ_c. This feature would greatly enhance

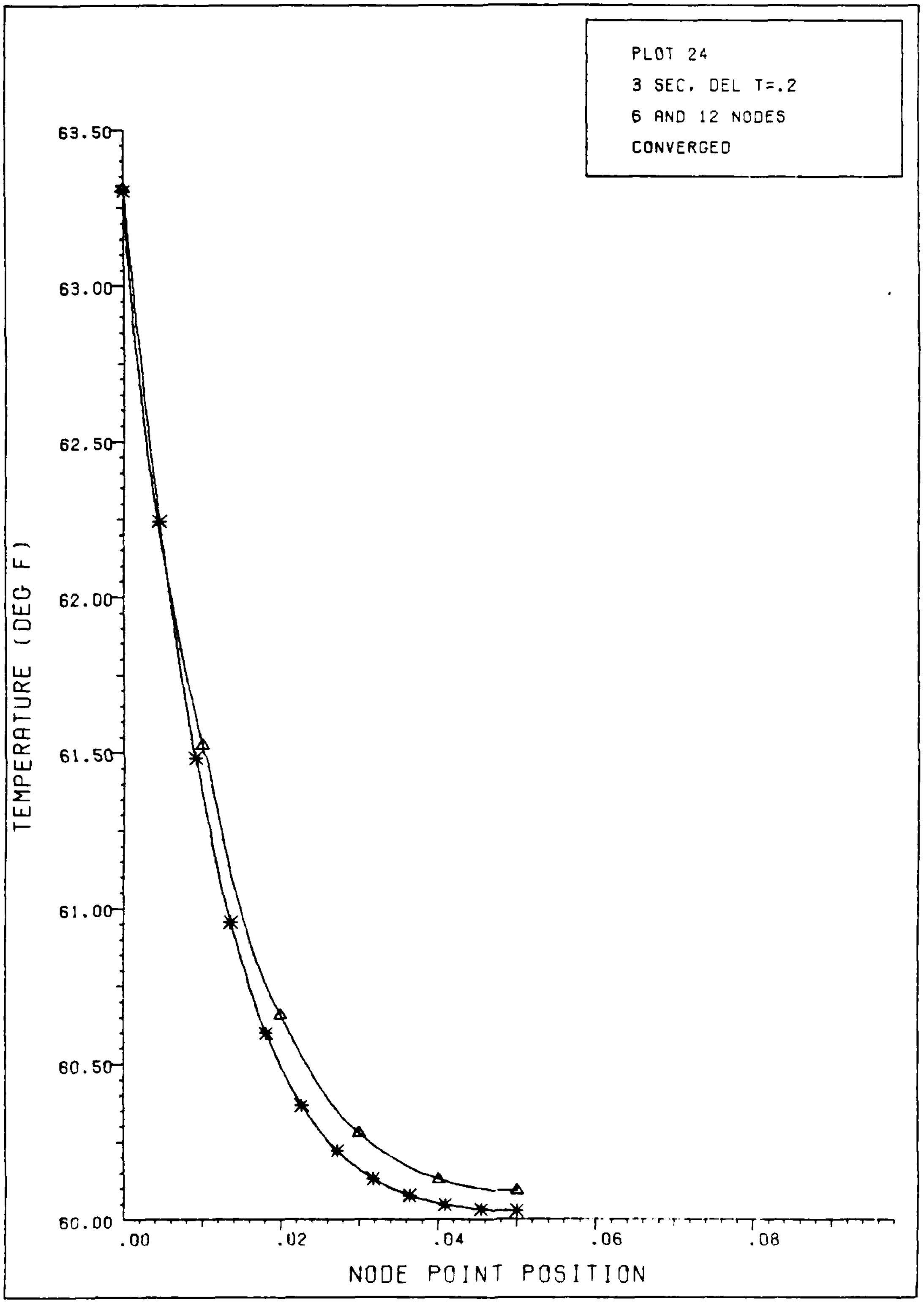


FIGURE 4.17 TEMP VS NODE POS.(6 NODES VS 12 NODES)

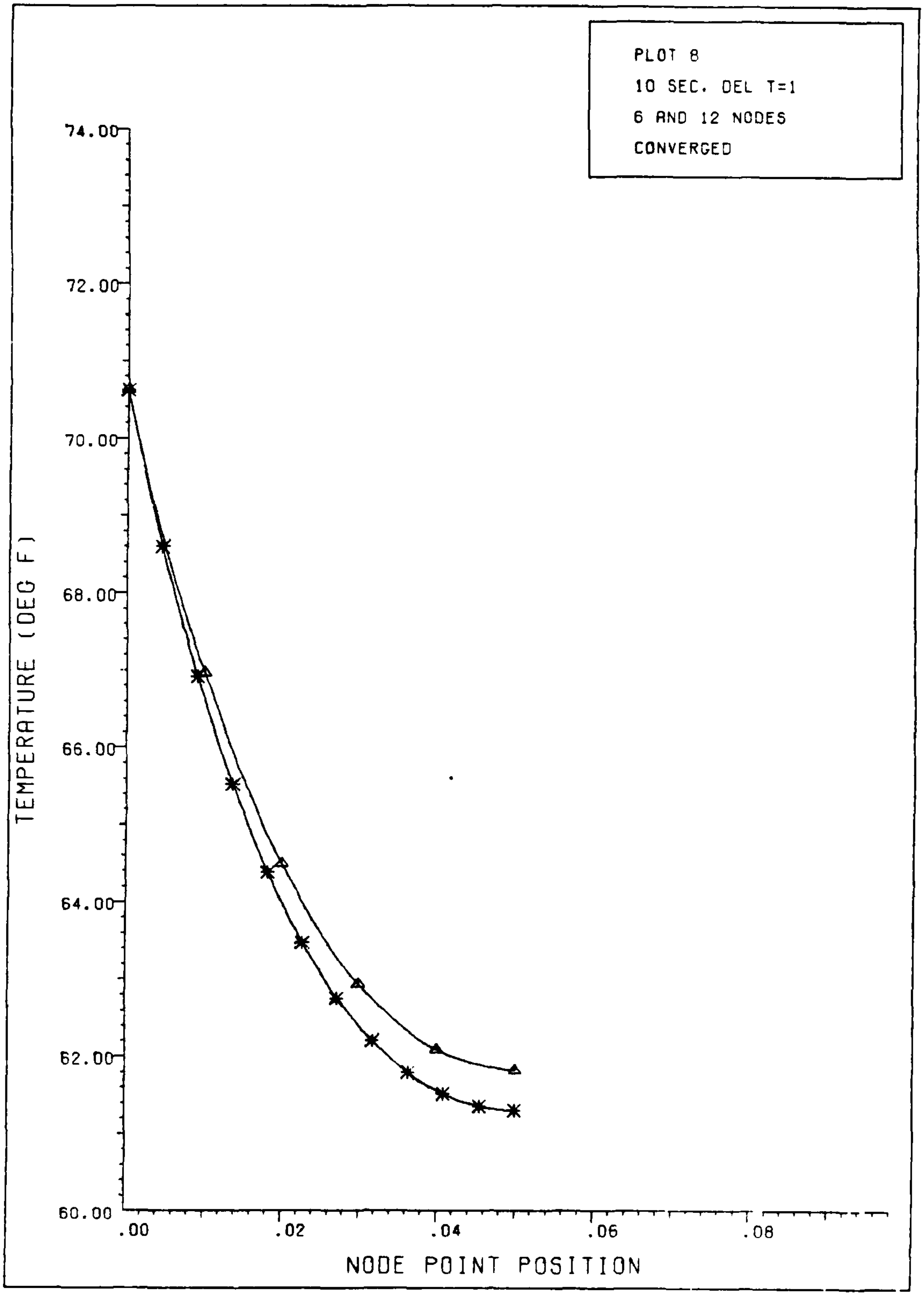


FIGURE 4.16 TEMP VS NODE POS.(6 NODES VS 12 NODES)

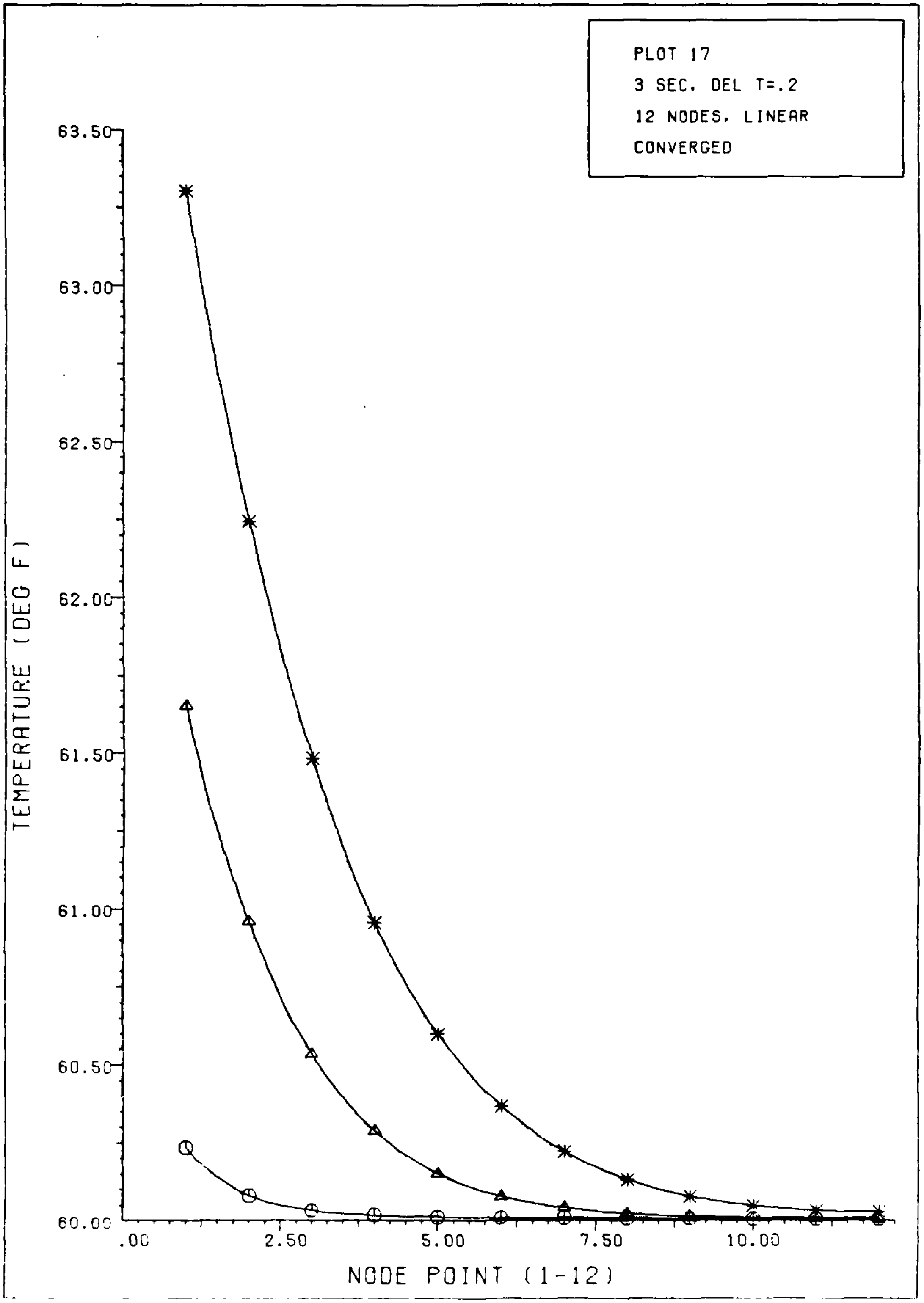


FIGURE 4.15 TEMP VS NODE POINT (.4, 1.6, 3.0 SEC.)

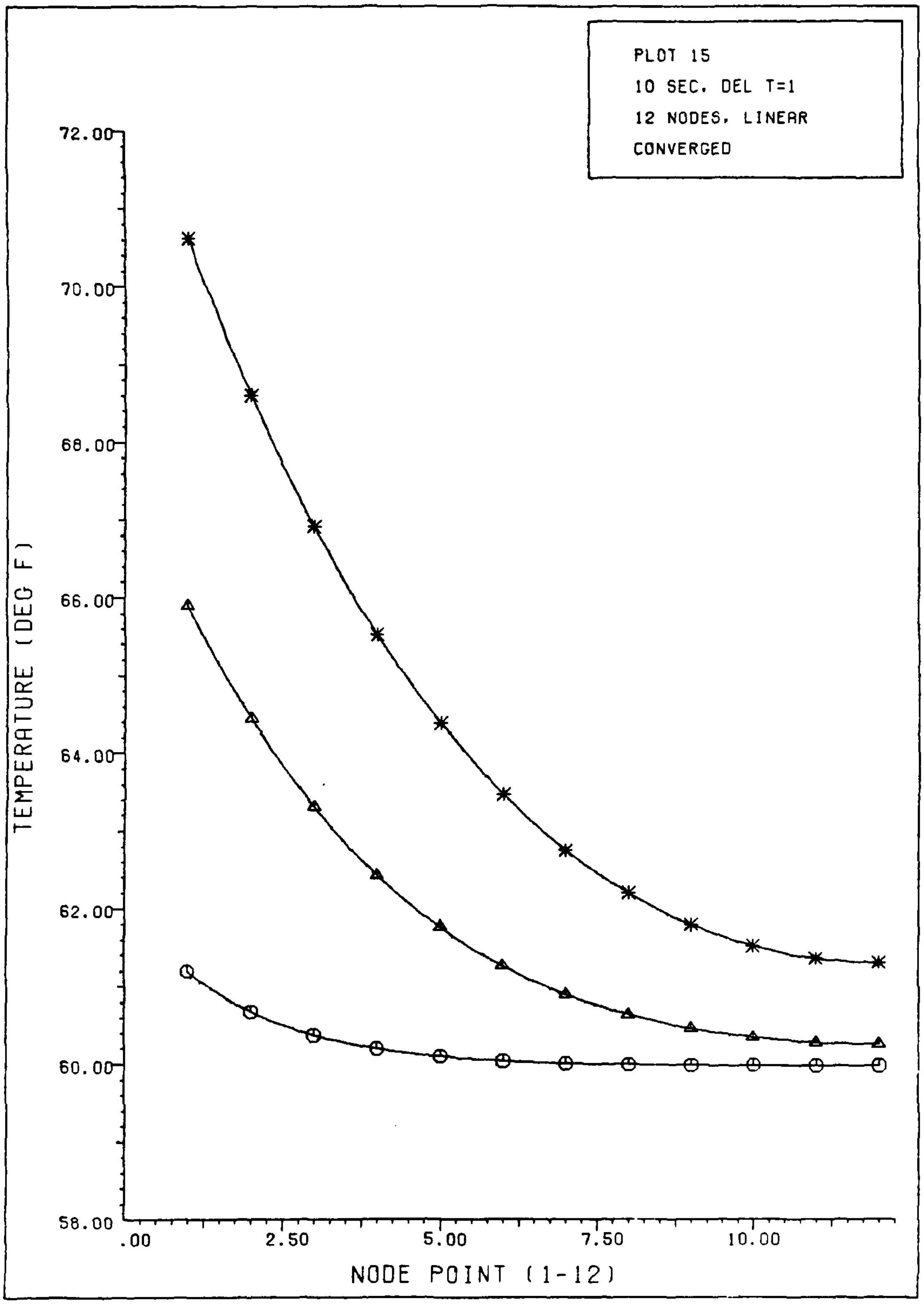


FIGURE 4.14 TEMP VS NODE POINT HISTORY(2,6,10 SEC)

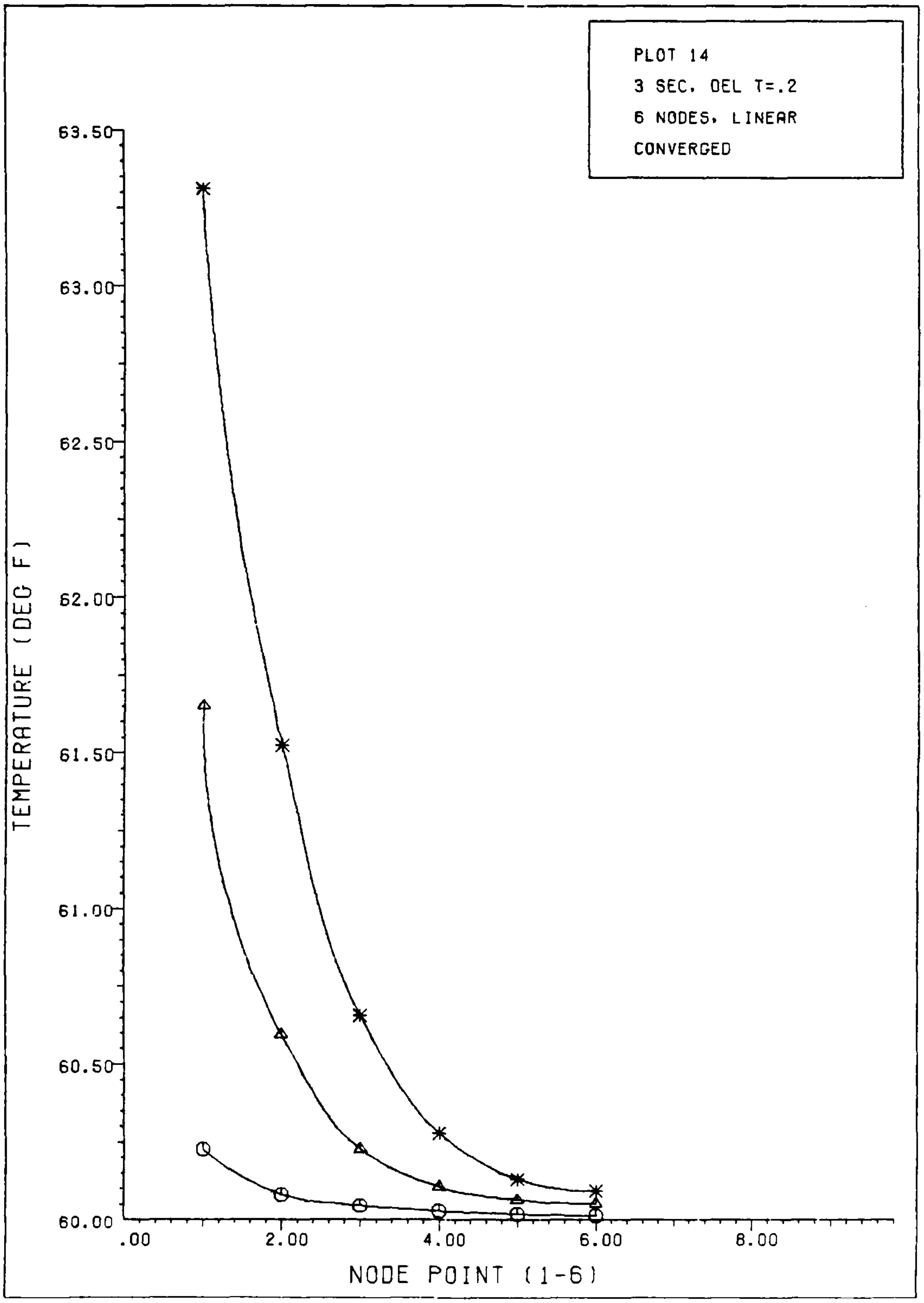


FIGURE 4.13 TEMP VS NODE POINT (.4, 1.6, 3.0 SEC)

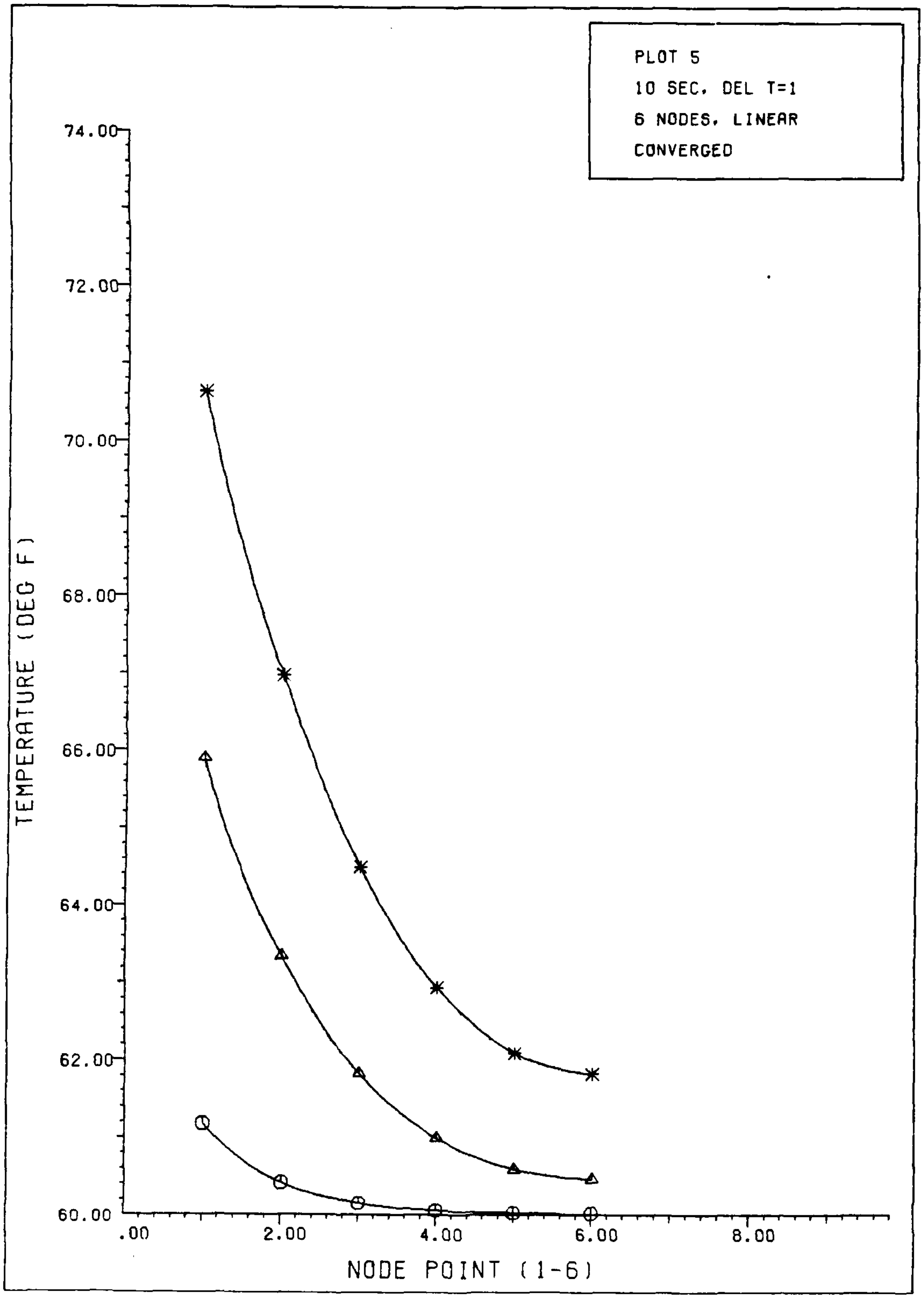


FIGURE 4.12 TEMP VS NODE POINT HISTORY(2,6,10 SEC)

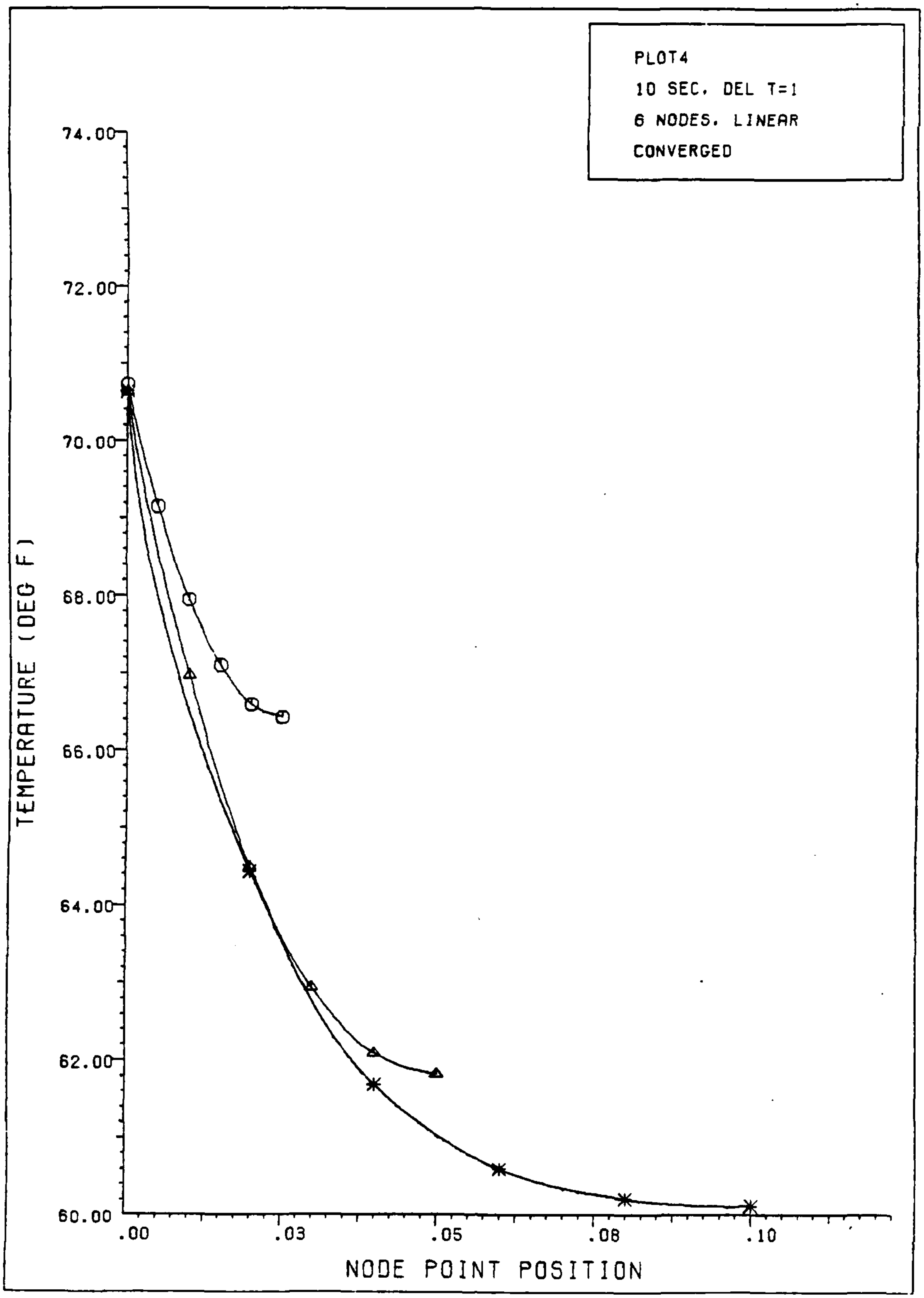


FIGURE 4.11 TEMP VS POS.(T/C LENGTH=.1,.05,.025FT)

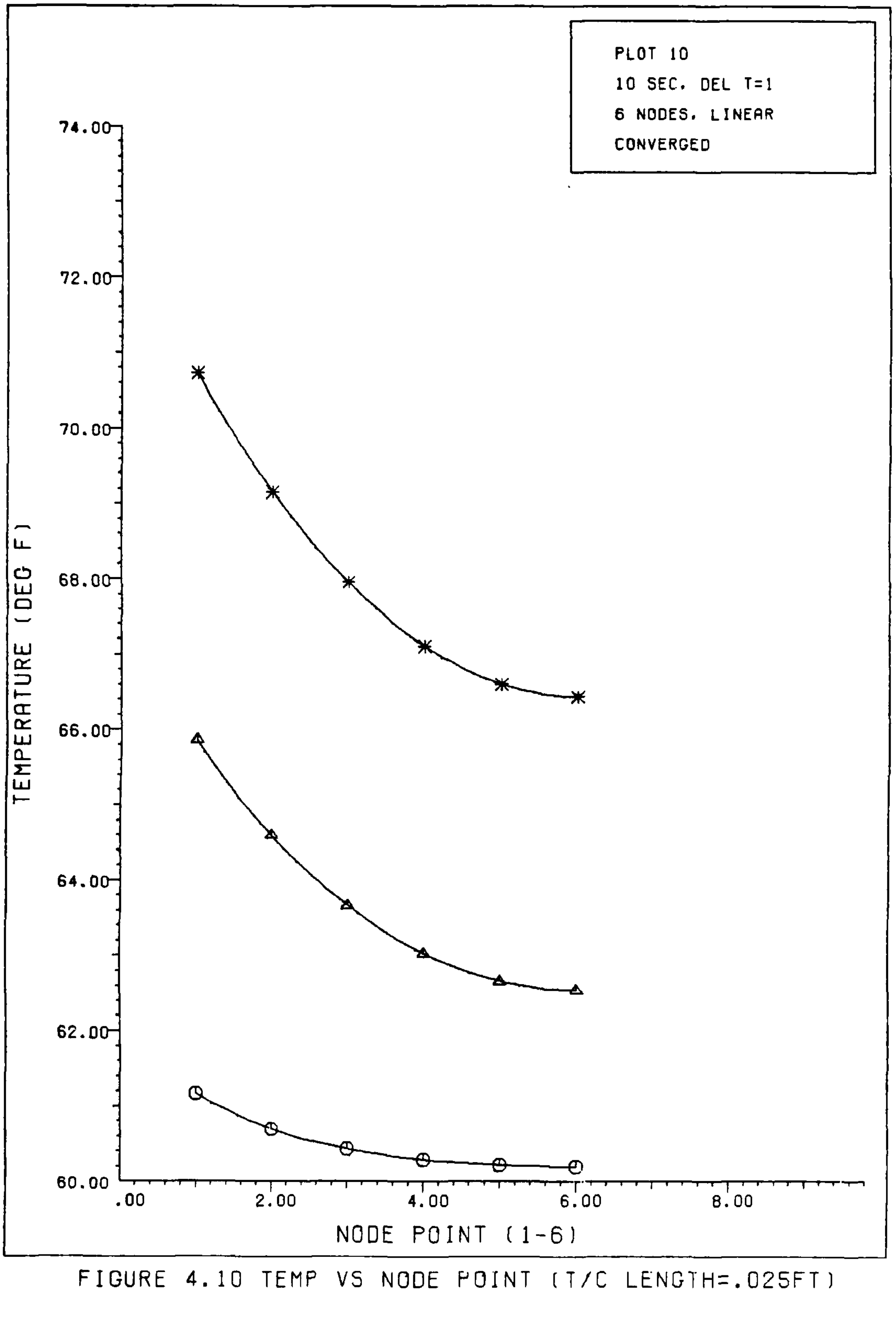


FIGURE 4.10 TEMP VS NODE POINT (T/C LENGTH=.025FT)

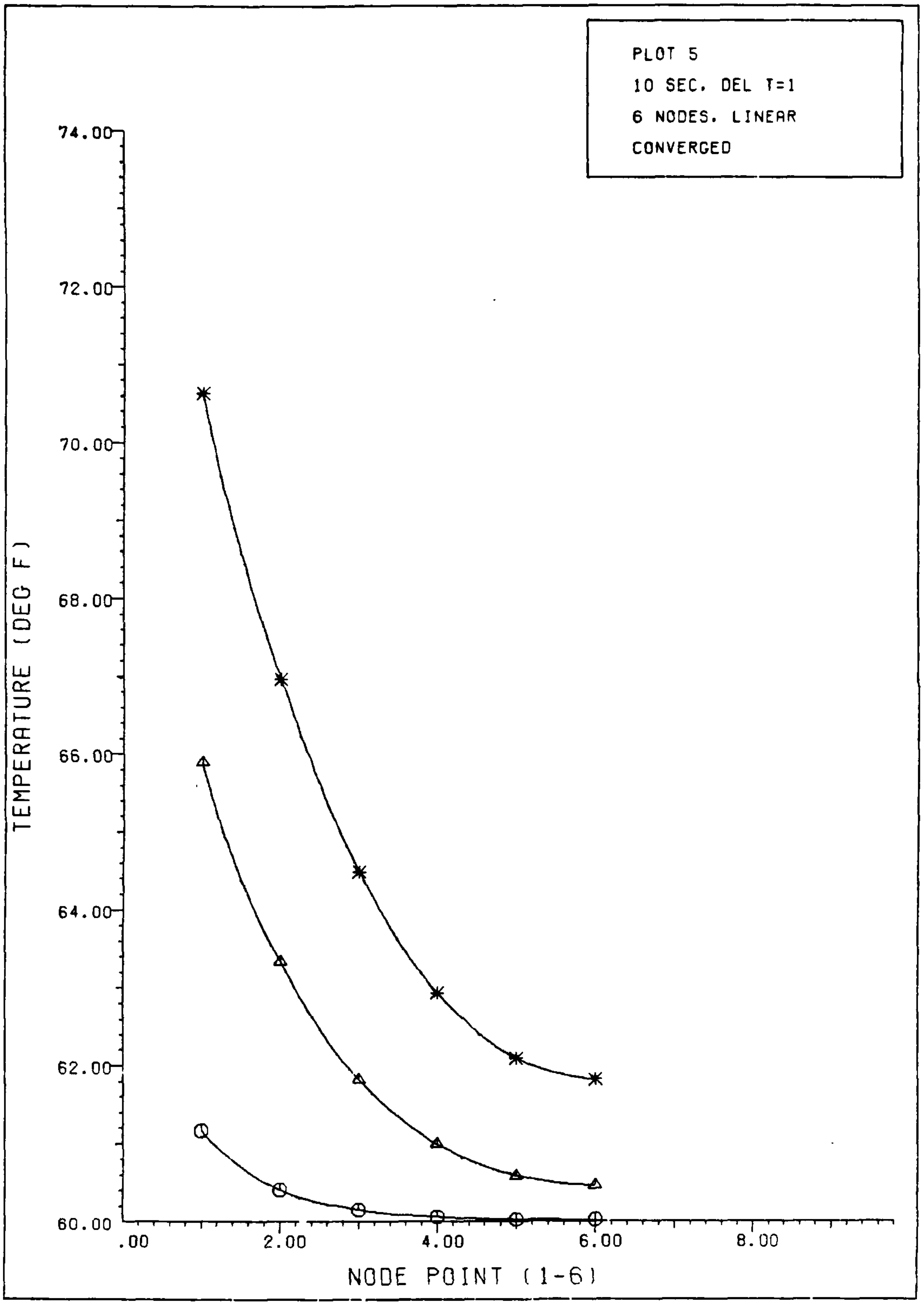


FIGURE 4.9 TEMP VS NODE POINT (T/C LENGTH= .05 FT)

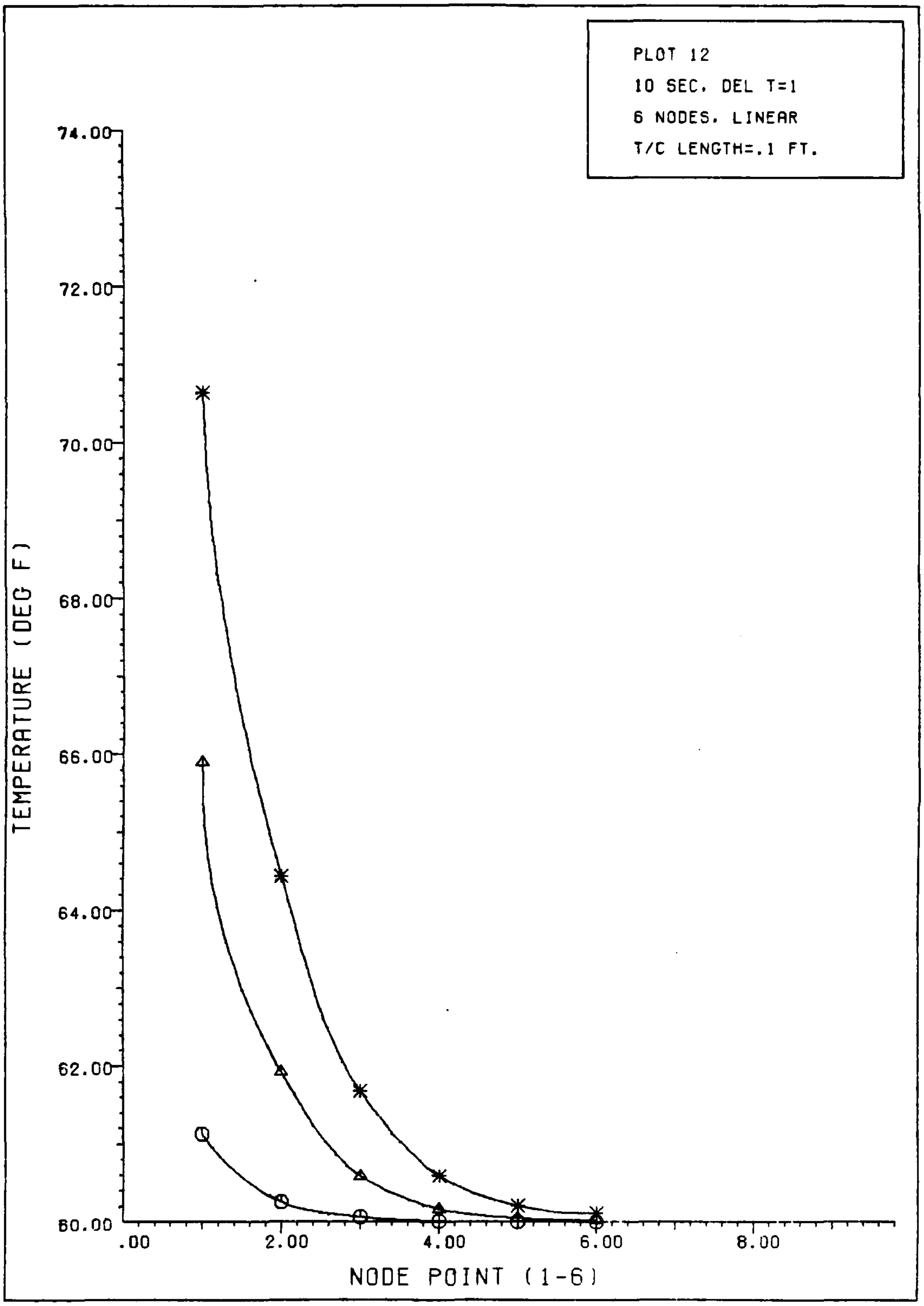


FIGURE 4.8 TEMP VS NODE POINT (T/C LENGTH= .1 FT.)

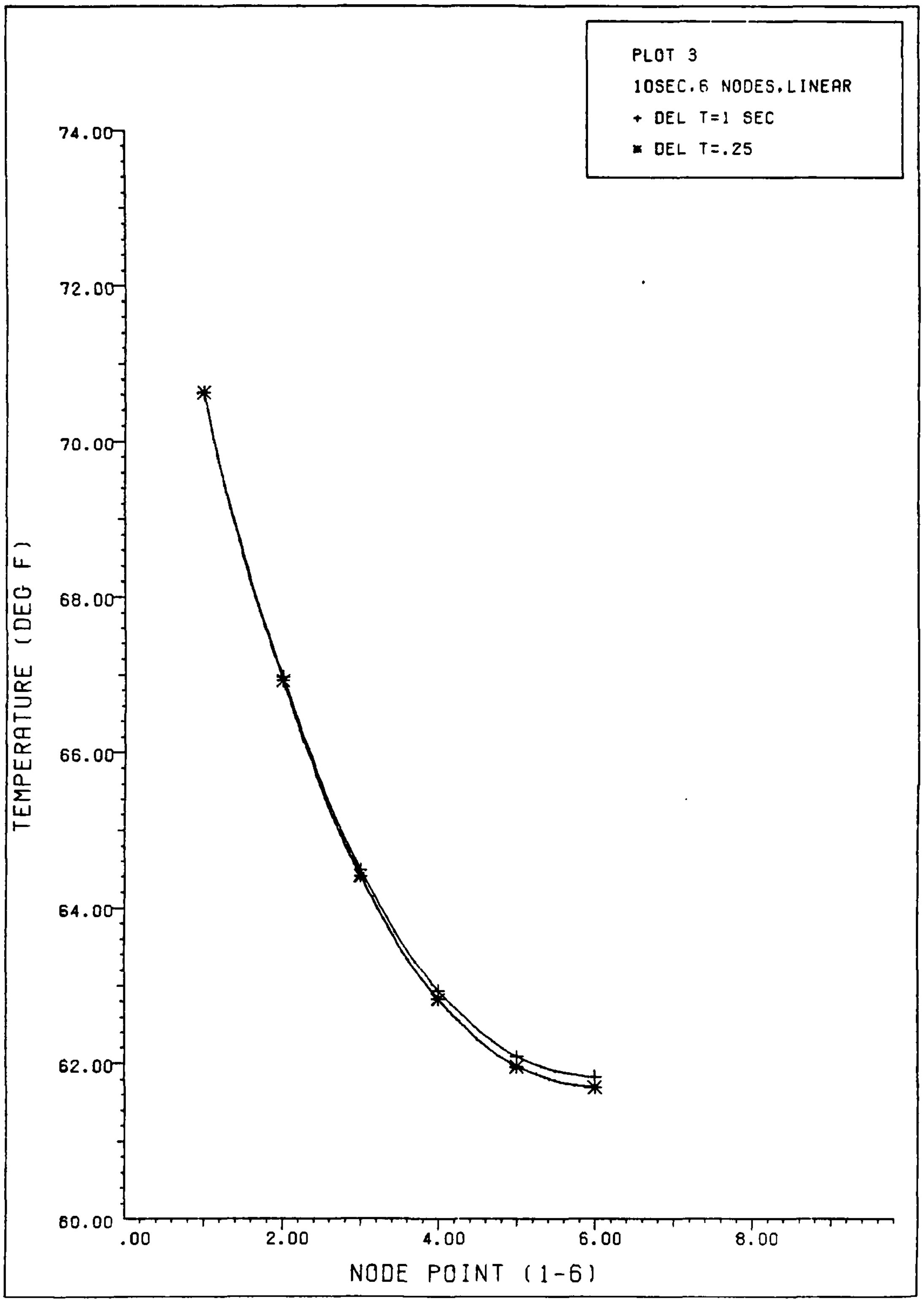


FIGURE 4.7 TEMP VS NODE POINT(DEL T=1 AND .25 SEC)

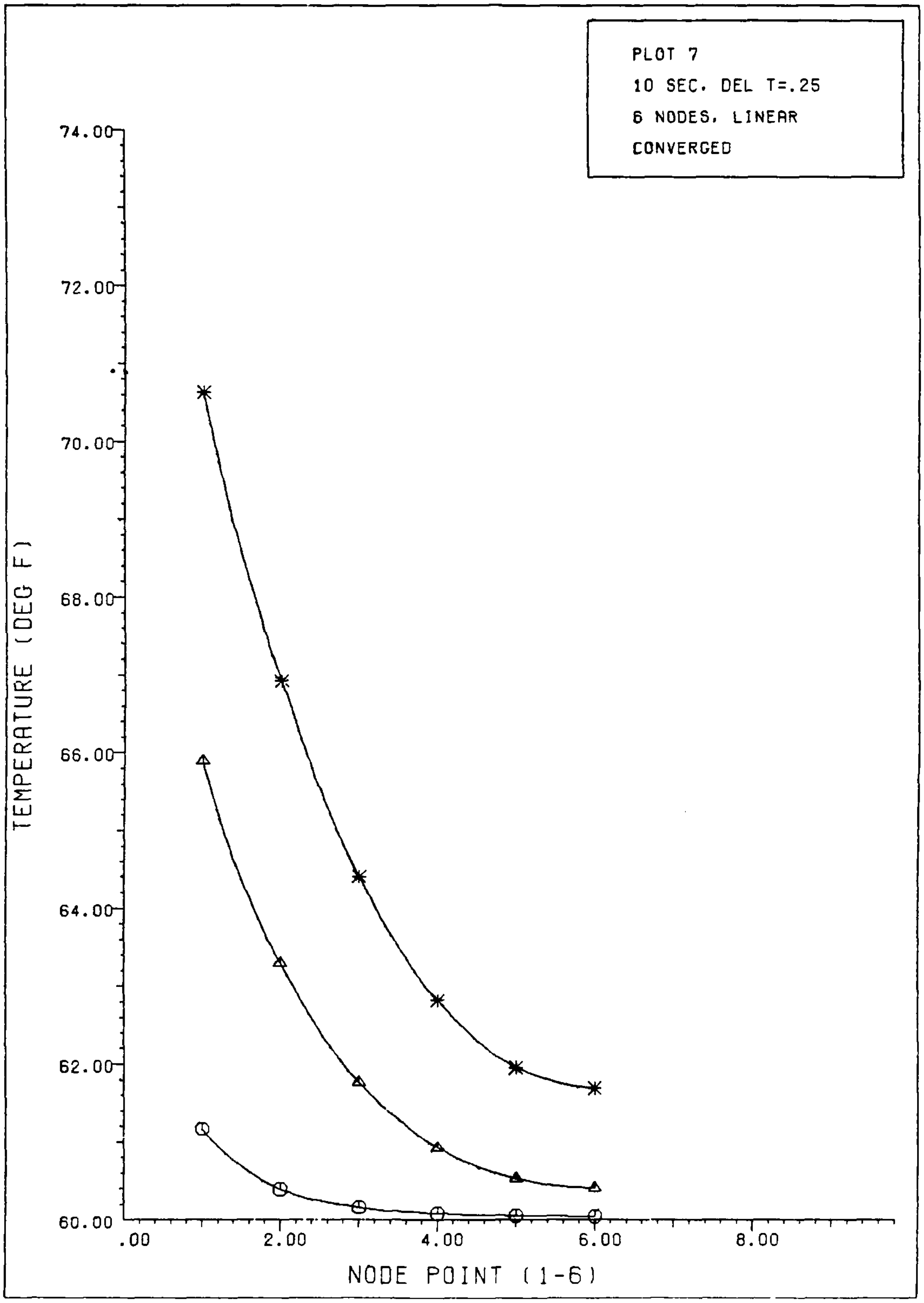
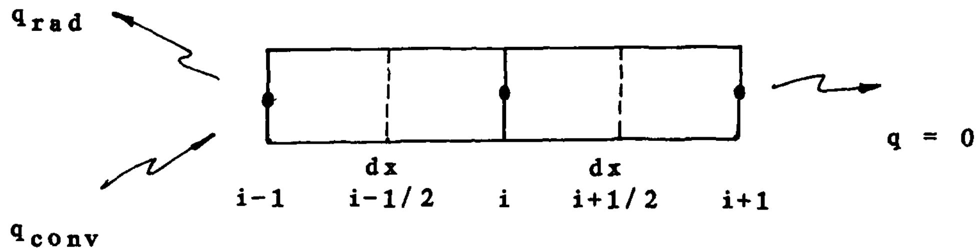


FIGURE 4.6 TEMP VS NODE POINT HISTORY (2,6,10 SEC)

APPENDIX A

Derivation of equations using a one-dimensional energy balance formulation are given as follows,



Energy in the left face $= -k\partial T/\partial x = q_1$

Energy generated within the element $= q\ dx = 0$

Change in internal energy $= \rho c(\partial T/\partial \tau)dx$

Energy out of right face $= -k(\partial T/\partial x)_{x+dx}$

$$= -[k\frac{\partial T}{\partial x} + \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)dx]$$

Then, combining the above and using Fourier's Law of Heat Conduction, ie,

energy in left face + energy within the element = change in internal energy + energy out right face

yields,

$$-k\frac{\partial T}{\partial x} + q dx = \rho c\frac{\partial T}{\partial \tau}dx - \left[k\frac{\partial T}{\partial x} + \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)dx\right] \quad \text{(A-1)}$$

or,

$$\rho c\frac{\partial T}{\partial \tau} = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) \quad \text{(A-2)}$$

or, replacing T by U,

$$\rho c U_t = (kU_x)_x \tag{A-3}$$

or,

$$\rho c U_t = \left[\frac{k_{i-1/2}}{\Delta x_{i-1/2}} (U_{i-1}{}^n - U_i{}^n) - \frac{k_{i+1/2}}{\Delta x_{i+1/2}} (U_i{}^n - U_{i+1}{}^n)\right]\frac{1}{\Delta x} \tag{A-4}$$

where ΔX may be written as,

$$\Delta x = \frac{\Delta x_{i+1/2} + \Delta x_{i-1/2}}{2}$$

then, writing the time gradient in first order backward difference form and expanding yields,

$$\rho c \frac{U_i{}^n - U_{i-1}{}^n}{\Delta t} = \frac{2k_{i-1/2}}{\Delta x_{i-1/2}(\Delta x_{i+1/2} + \Delta x_{i-1/2})} U_{i-1}{}^n$$

$$- \frac{2k_{i-1/2}}{\Delta x_{i-1/2}(\Delta x_{i+1/2} + \Delta x_{i-1/2})} U_i{}^n$$

$$- \frac{2k_{i+1/2}}{\Delta x_{i+1/2}(\Delta x_{i+1/2} + \Delta x_{i-1/2})} U_i{}^n$$

$$+ \frac{2k_{i+1/2}}{\Delta x_{i+1/2}(\Delta x_{i+1/2} + \Delta x_{i-1/2})} U_{i+1}{}^n \tag{A-5}$$

then, specifying equal spacing for each node point,

$$\Delta x_{i+1/2} = \Delta x_{i-1/2}$$

and the equation becomes,

$$\rho c \frac{U_i^{\,n} - U_i^{\,n-1}}{\Delta t} = \frac{k_{i-1/2}}{\Delta x^2} U_{i-1}^{\,n} - \left(\frac{k_{i-1/2}}{\Delta x^2} + \frac{k_{i+1/2}}{\Delta x^2}\right) U_i^{\,n} + \frac{k_{i+1/2}}{\Delta x^2} U_{i+1}^{\,n} \tag{A-6}$$

Now, instead of estimating c and k directly, define two scaling parameters ϕ_c and ϕ_k such that c and k will remain constant. These two parameters are estimated by the HEATEST program.

$$\rho \phi_c c \Delta x \frac{U_i^{\,n} - U_i^{\,n-1}}{\Delta t} = \phi_k \frac{k_{i-1/2}}{\Delta x} U_{i-1}^{\,n} - \phi_k \left(\frac{k_{i-1/2}}{\Delta x} + \frac{k_{i+1/2}}{\Delta x}\right) U_i^{\,n} + \phi_k \frac{k_{i+1/2}}{\Delta x} U_{i+1}^{\,n} \tag{A-7}$$

This equation is applicable at all interior ($i \neq 1, i \neq i_{max}$) points.

For the back face, assuming a semi-infinite solid, $i = i_{max}$ and the equation becomes,

$$\frac{\rho \phi_c c \Delta x}{2} \frac{U_L^{\,n} - U_L^{\,n-1}}{\Delta t} = \frac{\phi_k k_{L-1/2}}{\Delta x} U_{L-1}^{\,n} - \frac{\phi_k k_{L-1/2}}{\Delta x} U_L^{\,n} \tag{A-8}$$

For the front face, $(i = 1)$, the effects of radiation away from and convection toward the solid surface must be accounted for.

The radiation is modeled using the Stefan-Boltzmann Law,

$$q = \varepsilon\sigma(U_1{}^4 - U_\infty{}^4) \tag{A-9}$$

where ε radiative emissivity
σ Stefan-Boltzmann constant
U Temperature ($^\circ$R)

The convective transfer of energy is modeled using Newton's Law of Cooling,

$$q = h(T_{aw} - T_w) \tag{A-10}$$

Non-dimensionalizing by a reference heat transfer coefficient, h_{ref} yields,

$$q = h_{bar} h_{ref}(T_{aw} - U_1) \tag{A-11}$$

where, h_{bar} = convective heat transfer coefficient ratio
T_{aw} = adiabatic wall temp of test article

The dependance of the heat transfer coefficient on parameters other than those included in the reference heat transfer coefficient are summarized by the static transfer relation or heat transfer coefficient ratio, h/h_{ref}. Here, the ratio is assumed to be piecewise linear with respect to angle of attack as derived from Lagrange Interpolation Theory (Ref 6),

$$h_{bar} = h/h_{ref} = [h_0 + h_{\alpha 1}(\alpha - \alpha_1) + h_{\alpha 2}(\alpha - \alpha_2)] \qquad \text{(A-12)}$$

where h_0 is the magnitude of the heat transfer coefficient, h, at the reference condition specified at α_1. Combining Equations A-7, A-9, A-11, and A-12 and evaluating at node one yields,

$$\frac{\rho \phi_c c \Delta x}{2} \frac{U_1^n - U_1^{n-1}}{\Delta t} = \frac{-\phi_k k_{k+1/2}}{\Delta x} U_1^n$$
$$+ \frac{\phi_k k_{1+1/2}}{\Delta x} U_2^n - \varepsilon\sigma[(U_1^n)^4 - (U_\infty^n)^4]$$
$$+ [h_0 + h_{\alpha 1}(\alpha - \alpha_1) + h_{\alpha 2}(\alpha - \alpha_2)] h_{ref} (T_{aw} - U_1^n)$$
$$\text{(A-13)}$$

Using the quasi-linearization as developed in Equation 2-6, the resultant form for determining the temperature time history at each node point is given in Equations 2-7 and 2-8.

The matrix form for the equations may be found after defining the following,

$$RCX_i = \rho \phi_c c \Delta x \qquad RCX_1 = \frac{RCX_1}{2} \qquad RCX_L = \frac{RCX_L}{2}$$

$$RM_i = \frac{\phi_k k_{i-1/2}}{\Delta x} \qquad RM_1 = 0$$

$$RP_i = \frac{\phi_k k_{i+1/2}}{\Delta x} \qquad RP_L = 0$$

$$BBB_1 = \frac{RCX_1}{\Delta t} + RM_1 + RP_1 + 4\varepsilon\sigma(U_1^{n,s})^3 + h_{bar}h_{ref}$$

$$BBB_i = \frac{RCX_i}{\Delta t} + RM_i + RP_i$$

(A-14)

Then, using Equations A-14 in Equations 2-7 and 2-8 yields the matrix form of Equation 2-9,

$$[A]\{U_i^n\} + \{b\} = 0 \qquad (2\text{-}9)$$

where,

$$[A] = \begin{bmatrix} -1 & RP_i/BBB_i & & 0 \\ RM_i/BBB_i & \ddots & \ddots & \\ & \ddots & \ddots & RP_i/BBB_i \\ 0 & & RM_i/BBB_i & -1 \end{bmatrix}$$

(A-15)

and,

$$\{b\} = \left\{ \begin{array}{l} \dfrac{\varepsilon\sigma[(3U_1^{n,s})^4 + U_\infty^4] + h_{bar}h_{ref}T_{aw} + \frac{RCX_1}{\Delta t}U_1^{n-1}}{BBB_1} \\ \dfrac{RCX_i}{\Delta t}U_i^{n-1} \Big/ BBB_i \\ \\ \dfrac{RCX_L}{\Delta t}U_L^{n-1} \Big/ BBB_L \end{array} \right\}$$

(A-16)

APPENDIX B

The derivation of the sensitivity equations. The derivative of Equation 2-7 with respect to each parameter yields equations from which the HEATEST program propogates the sensitivity. A vector of parameters is formed and the sensitivity notation is as shown,

$$\theta = [h_0,\ h_{\alpha 1},\ h_{\alpha 2},\ \phi_c,\ \phi_k)]^T \qquad S_{i,k} = \frac{\partial U}{\partial \theta_k}$$

parameter no

node point

Defining, $h_{bar} = [h_0 + h_{\alpha 1}(\alpha - \alpha 1) + (h_{\alpha 2}(\alpha - \alpha 2)]$, the sensitivity equations at node one are,

$$\theta_1: \quad \frac{\rho \phi_c c \Delta x}{2} \frac{S_{1,1}{}^{n} - S_{1,1}{}^{n-1}}{\Delta t} = \frac{-\phi_k(k_{i+1/2})}{\Delta x} S_{1,1}{}^{n}$$

$$+ \frac{\phi_k(k_{i+1/2})}{\Delta x} S_{2,1}{}^{n} - 4\varepsilon\sigma(U_1{}^{n})^3\ S_{1,1}{}^{n} + h_{ref}(T_{aw} - U_1{}^{n})$$

$$- S_{1,1}{}^{n} h_{ref} h_{bar} \tag{B-1}$$

$$\theta_2: \quad \frac{\rho \phi_c c \Delta x}{2} \frac{S_{1,2}{}^{n} - S_{1,2}{}^{n-1}}{\Delta t} = \frac{-\phi_k(k_{i+1/2})}{\Delta x} S_{1,2}{}^{n}$$

$$+ \frac{\phi_k(k_{i+1/2})}{\Delta x} S_{2,2}{}^{n} - 4\varepsilon\sigma(U_1{}^{n})^3\ S_{1,2}{}^{n}$$

$$+ (\alpha - \alpha 1) h_{ref}(T_{aw} - U_1{}^{n}) - S_{1,2}{}^{n} h_{ref} h_{bar} \tag{B-2}$$

$$\theta_3: \quad \frac{\rho\phi_c c\Delta x}{2}\,\frac{S_{1,3}{}^{n}-S_{1,3}{}^{n-1}}{\Delta t} = \frac{-\phi_k(k_{i+1/2})}{\Delta x}\,S_{1,3}{}^{n}$$

$$+\ \frac{\phi_k(k_{i+1/2})}{\Delta x}\,S_{2,3}{}^{n} - 4\varepsilon\sigma(U_1{}^{n})^3\,S_{1,3}{}^{n}$$

$$+\ (\alpha-\alpha 2)h_{ref}(T_{aw}-U_1{}^{n}) - S_{1,3}{}^{n}h_{ref}h_{bar}$$

(B-3)

$$\theta_4: \quad \frac{S_{1,4}{}^{n}-S_{1,4}{}^{n-1}}{\Delta t} = \frac{-2\phi_k k_{i+1/2}}{\rho\phi_c c\Delta x^2}\,S_{1,4}{}^{n} + \frac{2\phi_k k_{1+1/2}}{\rho\phi_c{}^2 c\Delta x^2}\,U_1{}^{n}$$

$$+\ \frac{2\phi_k k_{1+1/2}}{\rho\phi_c c\Delta x^2}\,S_{1,4}{}^{n} - \frac{2\phi_k k_{1+1/2}}{\rho\phi_c{}^2 c\Delta x^2}\,U_2{}^{n}$$

$$-\ \frac{8\varepsilon\sigma(U_1{}^{n})^3}{\rho\phi_c c\Delta x}\,S_{1,4}{}^{n} + \frac{2\varepsilon\sigma(U_1{}^4-U_\phi{}^4)}{\rho\phi_c{}^2 c\Delta x}$$

$$-\ \frac{2h_{bar}h_{ref}}{\rho\phi_c c\Delta x}\,S_{1,4}{}^{n} - \frac{2h_{bar}h_{ref}(T_{aw}-U_1)}{\rho\phi_c{}^2 c\Delta x}$$

(B-4)

$$\theta_5: \quad \frac{\rho\phi_c c\Delta x}{2}\,\frac{S_{1,5}{}^{n}-S_{1,5}{}^{n-1}}{\Delta t} = \frac{-\phi_k k_{1+1/2}}{\Delta x}\,S_{1,5}{}^{n}$$

$$-\ \frac{k_{1+1/2}}{\Delta x}\,U_1{}^{n} + \frac{\phi_k k_{1+1/2}}{\Delta x}\,S_{2,5}{}^{n} + \frac{k_{1+1/2}}{\Delta x}\,U_2{}^{n}$$

$$-\ 4\varepsilon\sigma U_1{}^3 S_{1,5} - S_{1,5}h_{ref}h_{bar}$$

(B-5)

The sensitivity equations at the interior node points are as

follows,

Θ_i , i = 1, 2, 3

$$\rho \phi_c c \Delta x \; \frac{S_{i,k}{}^{n} - S_{i,k}{}^{n-1}}{\Delta t} = \frac{\phi_k k_{i-1/2}}{\Delta x} S_{i-1,k}$$

$$- \frac{\phi_k (k_{i-1/2} + k_{i+1/2})}{\Delta x} S_{i,k}$$

$$+ \frac{\phi_k k_{i+1/2}}{\Delta x} S_{i+1,k}$$

(B-6)

Θ_4:

$$\frac{S_{i,4}{}^{n} - S_{i,4}{}^{n-1}}{\Delta t} = \frac{\phi_k k_{i-1/2}}{\rho \phi_c c \Delta x^2} S_{i-1,4}{}^{n} - \frac{\phi_k k_{i-1/2}}{\rho \phi_c{}^2 c \Delta x^2} U_{i-1}{}^{n}$$

$$- \frac{\phi_k}{\rho \phi_c c \Delta x} \; \frac{(k_{i-1/2} + k_{i+1/2})}{\Delta x} S_{i,4}{}^{n}$$

$$+ \frac{\phi_k}{\rho \phi_c{}^2 c \Delta x} \; \frac{(k_{i-1/2} + k_{i+1/2})}{\Delta x} U_i{}^{n} + \frac{\phi_k k_{i+1/2}}{\rho \phi_c c \Delta x^2} S_{i+1,4}{}^{n}$$

$$- \frac{\phi_k k_{i+1/2}}{\rho \phi_c{}^2 c \Delta x^2} U_{i+1}{}^{n}$$

(B-7)

Θ_5:

$$\rho \phi_c c \Delta x \; \frac{S_{i,5}{}^{n} - S_{i,5}{}^{n-1}}{\Delta t} = \frac{\phi_k k_{i-1/2}}{\Delta x} S_{i-1,5}$$

$$- \frac{\phi_k (k_{i-1/2} + k_{i+1/2})}{\Delta x} S_{i,5}$$

$$+ \frac{\phi_k k_{i+1/2}}{\Delta x} S_{i+1,5} + \frac{k_{i-1/2}}{\Delta x} U_{i-1}$$

$$- \frac{(k_{i-1/2} + k_{i+1/2})}{\Delta x} U_i + \frac{k_{i+1/2}}{\Delta x} U_{i+1}$$

(B-8)

The backface equations are of the same form as the node 1 equations without the convection and radiation terms.

If Equations A-14 are used to reduce the equations to the form of Equation 2-9, the sensitivity equations become,

$$[A']\{S_{i,k}\} + \{d_k\} = 0 \qquad \text{(B-9)}$$

where tha [A] matrix for the sensitivity equations is the same as the [A] matrix for the temperature equations, A-15. The {d} vectors for each parameter are listed as follows,

$$\{d\}1 = \left\{ \begin{array}{l} \left(h_{ref}(T_{aw}-U_1) + \frac{RCX_1}{\Delta t} S_{1,1}{}^{n-1}\right) / BBB_1 \\ \left(\frac{RCX_i}{\Delta t} S_{i,1}{}^{n-1}\right) / BBB_i \\ \\ \end{array} \right\}$$

$$\{d\}2 = \left\{ \begin{array}{l} \left((\alpha-\alpha 1) h_{ref}(T_{aw}-U_1) + \frac{RCX_1}{\Delta t} S_{1,2}{}^{n-1}\right) / BBB_1 \\ \left(\frac{RCX_i}{\Delta t} S_{i,2}{}^{n-1}\right) / BBB_i \\ \\ \end{array} \right\}$$

$$
\{d\}3 = \begin{Bmatrix} \left((\alpha-\alpha 2)h_{ref}(T_{aw}-U_1) + \frac{RCX_1}{\Delta t} S_{1,3}^{n-1}\right) \Big/ BBB_1 \\ \left(\frac{RCX_i}{\Delta t} S_{i,3}^{n-1}\right) \Big/ BBB_i \end{Bmatrix}
$$

$$
\{d\}4 = \begin{Bmatrix} \left(-RP_1\frac{(U_2-U_1)}{\phi_c} + \varepsilon\sigma\frac{(U_1^{\,4}-U_\infty^{\,4})}{\phi_c} - \frac{h_{bar}h_{ref}(T_{aw}-U_1)}{\phi_c} + \frac{RCX_1}{\Delta t} S_{i,4}^{n-1}\right) \Big/ BBB_1 \\ \left(-\frac{RM_i}{\phi_c}(U_{i-1}-U_i) + \frac{RP_i}{\phi_c}(U_i-U_{i+1}) + \frac{RCX_i}{\Delta t} S_{i,4}\right) \Big/ BBB_i \end{Bmatrix}
$$

$$
\{d\}5 = \begin{Bmatrix} \left(\frac{RP_1}{\phi_k}(U_2-U_1) + \frac{RCX_1}{\Delta t} S_{i,5}^{n-1}\right) \Big/ BBB_1 \\ \left(\frac{RM_i}{\phi_k}U_{i-1} - \frac{(RM_i+RP_i)}{\phi_k}U_i + \frac{RP_i}{\phi_k}U_{i+1} + \frac{RCX_i}{\Delta t} S_{i,5}^{n-1}\right) \Big/ BBB_i \end{Bmatrix} \tag{B-10}
$$

APPENDIX C

The HEATEST program follows.

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      PROGRAM HEATEST (INPUT,OUTPUT,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6=          OCT10       6
     *OUTPUT,TAPE9,TAPE10,TAPE12,TAPE13,TAPE21,TAPE30)                      OCT10       7
C                                                                           FHEATEST2   4
C  TAPE2 = CALCOMP PLOT FILE                                                FHEATEST2   5
C  TAPE3 = TEMP ICS                                                         FHEATEST2   6
C  TAPE4 = MANEUVER (BET) FILE                                              FHEATEST2   7
C  TAPE9 = TEMP IC OUTPUT FILE FROM ADAPTIVE FILTER MODE                    FHEATEST2   8
C  TAPE10 = ADDAPTIVE TEMPERATURE/STATE ESTIMATE TIME SERIES                FHEATEST2   9
C  TAPE12 = FORMATTED TIME SERIES FOR PLOT                                  FHEATEST2  10
C  TAPE13 = THERMOCOUPLE MEASUREMENTS                                       FHEATEST2  11
C                                                                           FHEATEST2  12
      REAL M1                                                               FHEATEST2  13
      REAL TW1                                                              UPOCT09    10
      LOGICAL IFICENT,IFPLOT,IFPRINT,IFXFLG                                 FHEATEST2  14
      COMMON /CFLAG/IFICENT,IFPLOT,IFPRINT                                  UPOCT09    11
      COMMON/CTCMMT/NTCT                                                    UPAUG16     9
      COMMON/COSP/NPTSS,USI(6),PHI(6,6),NPT,PC(6,6),RR,                     UPAUG16     6
     &QD(6,6),QDT(6,6),QUE(6,6),A(6,6),RCX(6),RP(6),RM(6)                   UPOCT09     5
      COMMON/CSENS/SUSI(6,5),UM1(6)                                         UPAUG16     7
      LOGICAL FREAD(13)                                                     FHEATEST2  21
      DIMENSION CI(5)                                                       UPAUG16    10
      LOGICAL FAUTO                                                         CPARAM      2
      DIMENSION FAUTO(7)                                                    OCT10       2
      DIMENSION QP(5)                                                       UPAUG16     1
      COMMON/CPARAM/HO,HALF(2),PHIC,PHIK,ZP,Z,ALPH(2),KA,S(5),              UPAUG16     2
     &CIF(5,5),KAF,IFX(5),ACC(5),IFXSUM,NPAR,DALPH(2)                       UPAUG16     3
      EQUIVALENCE (HO,QP(1))                                                UPAUG16     4
      COMMON/COMTUN/T,TAW1,ALPHA,H,V,RHO,P,TEMP,C,TRAD,RHOG,                OCT10       3
     &TO,TSINK,XFT,DEL,PDEL                                                 COMTUN      3
      COMMON/CHEAT/Q,TS,QREF,TW,M1,RENS,HBAR,HREF                           UPAUG16    12
      COMMON/ICTPS2/TINIT(1),ERALOW,E                                       UPAUG16    13
      COMMON/CTIME/TSTART,TSTOP,DTPENT,NRPITER,ITPRAM                       FHEATEST2  30
      COMMON/CDX/DX(1)                                                      UPAUG16    14
       COMMON /CONST/ XP1(13)                                               HAROLD      1
      COMMON/CPARAM2/AVERROR,EQUI,UMEAS                                     UPOCT09    12
      COMMON /CFPLOT/IPLOT,TSCALE,TMIN,TAXL,YSCALE,YMIN,YAXL,ASCALE,AMIN FHEATEST2  34
     *,AAXL                                                                 FHEATEST2  35
      COMMON /CSMTH/ UICSM(6),PICSM(6,6),UAP(6),PAP(6,6),                   UPOCT09     1
     &SMIC,TSMTH,W(6,6)                                                     UPOCT09     2
      LOGICAL SMIC                                                          FCSMTH      4
      COMMON/CKF/K(6),S1(6),J1(6,6),TC(2),NODES(2),KFOPT                    UPAUG16     8
      REAL K,J1                                                             FDKF        3
      COMMON/CPC/NPTPC                                                      FHEATEST2  37
      DIMENSION FLT1(2),FLT2(1)                                             UPAUG16    15
      EQUIVALENCE (TAW1,FLT1(1))                                            UPAUG16    16
      EQUIVALENCE (REF,ALPH)                                                UPAUG16    17
      EQUIVALENCE (TW1,TC(1))                                               UPAUG16    18
      DIMENSION TITLE(10)                                                   FHEATEST2  41
      DIMENSION REMARK(10)                                                  FHEATEST2  42
      DIMENSION PLAB1(12),PLAB2(12),NUM(12)                                 FHEATEST2  43
      DIMENSION VAR(13)                                                     HAROLD      2
      DIMENSION PU(2),PE(2),PUF(2),PEF(2)                                   UPAUG16    19
      DIMENSION READ(13)                                                    FHEATEST2  45
      DATA NPAR /5/                                                         UPAUG16    20
      DATA CLAB /1HC/                                                       FHEATEST2  47
```

```
            DATA LABELX/5HLABEL/                                        FHEATEST2  48
            DATA DTP/0/                                                 FHEATEST2  49
            DATA ERALOW/.5/                                             FHEATEST2  50
            DATA KA,KAF,IPE/1,1,1/                                      FHEATEST2  52
            DATA THDEP/.05/                                             OCT30       1
            DATA NPTSS/6/                                               UPSEP19     2
            DATA TRAD/0./                                               UPSEP19     3
      C                                                                 FHEATEST2  54
      C  READ INPUT DECK DATA                                           FHEATEST2  55
      C                                                                 FHEATEST2  56
            CALL INPUT                                                  FHEATEST2  57
      C                                                                 FHEATEST2  58
      C  NODE POINT STRUCTURE                                           FHEATEST2  59
      C                                                                 FHEATEST2  60
            DX(1)=THDEP/NPTSS                                           UPSEP19     4
            NODES(1)=1                                                  OCT24       1
            NODES(2)=NPTSS                                              OCT24       2
      C                                                                 FHEATEST2 120
      C                                                                 FHEATEST2 121
      C  ENTER OUTER/PARAMETER ESTIMATION ITERATION LOOP                FHEATEST2 122
      C                                                                 FHEATEST2 123
            READ(5,3990)ITRJSK,ITCSK,NPTPC,IIC                          FHEATEST2 124
       3990 FORMAT(8(8X,I2))                                            FHEATEST2 125
            READ(5,4001)INTERV,IFXFLG                                   FHEATEST2 126
       4001 FORMAT(6X,I4,9X,L1,10X,7L1)                                 FHEATEST2 127
            DELS=(TSTOP-TSTART)/INTERV                                  FHEATEST2 128
            TSTOPF=TSTOP                                                FHEATEST2 129
       4      TSTOP=TSTART+DELS                                         FHEATEST2 130
            IF(IFXFLG)THEN                                              FHEATEST2 131
            READ(5,4011,END=4034)NRPITER,IFX,FAUTO,TSTOP,KFOPT,NPTPC    FHEATEST2 132
       4011 FORMAT(4X,I1,5X,16I1,10X,7L1/8X,F10.5,5X,I1,8X,I2)          FHEATEST2 133
            READ(5,4012,END=4034)FREAD,READ                             FHEATEST2 134
       4012 FORMAT(3X,13L1,13F8.4)                                      FHEATEST2 135
            DO 4016 II=1,5                                              UPSEP24     1
      4016  IF(FREAD(II))QP(II)=READ(II)                                UPSEP24     2
            READ(5,4013,END=4034)FREAD,READ                             FHEATEST2 138
            DO 4017 II=1,2                                              UPSEP19     5
       4017 IF(FREAD(II))ALPH(II)=READ(II)                              UPSEP19     6
       4013 FORMAT(3X,7L1,7F10.5)                                       FHEATEST2 141
            GO TO 4033                                                  FHEATEST2 142
       4034 IFXFLG=.FALSE.                                              FHEATEST2 143
            TSTOP=TSTOPF                                                FHEATEST2 144
       4033 CONTINUE                                                    FHEATEST2 145
            END IF                                                      FHEATEST2 146
            IF(TSTOP.GT.TSTOPF)TSTOP=TSTOPF                             FHEATEST2 147
            IFXSUM=0                                                    FHEATEST2 149
            DO 30 I=1,NPAR                                              FHEATEST2 150
       30   IFXSUM=IFXSUM+IFX(I)                                        FHEATEST2 151
            NRPIT=NRPITER+1                                             FHEATEST2 152
            DO 198 ITPRAM=1,NRPIT                                       FHEATEST2 153
            KA=1                                                        FHEATEST2 154
            KAF=1                                                       FHEATEST2 155
            REWIND 3                                                    FHEATEST2 156
            REWIND 4                                                    FHEATEST2 157
            REWIND 10                                                   FHEATEST2 158
            REWIND 13                                                   FHEATEST2 159
            CALL ZERO(S,NPAR,1)                                         FHEATEST2 160
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

                SUBROUTINE TPS3(DTT)                                                    UPAUG1      11
                COMMON /CTCMMT/NTCT                                                     UPAUG1      12
                COMMON/COMTUN/T,TAW1,ALPHA,H,V,RHO,P,TEMP,C,TRAD,RHOG,                  OCT10        3
               &TO,TSINK,XFT,DEL,PDEL                                                   COMTUN       3
                COMMON /CHEAT/Q,TS,QREF,TW,M1,RENS,HBAR,HREF                            UPAUG1      14
                COMMON/COSP/NPTSS,USI(6),PHI(6,6),NPT,PC(6,6),RR,                       UPAUG16      6
               &QD(6,6),QDT(6,6),QUE(6,6),A(6,6),RCX(6),RP(6),RM(6)                     UPOCT09      5
                COMMON/CSENS/SUSI(6,5),UM1(6)                                           UPAUG16      7
                COMMON /ICTPS2/TINIT(1),ERALOW,E                                        UPAUG1      15
                COMMON /CDX/DX(1)                                                       UPAUG15      3
                REAL M1                                                                 FTPS3       16
                DIMENSION AAAA(6),CCCC(6),DDDD(6),G(6),W(6)                             UPOCT09      8
                EQUIVALENCE (QD(1,1),AAAA(1)),(QD(1,2),CCCC(1)),(QD(1,3),DDDD(1)),      FTPS3       19
               &(QD(1,5),G(1)),(QDT(1,1),W(1))                                          FTPS3       20
                DATA SIG/4.761E-13/                                                     UPAUG1      16
                DATA MIT/2/                                                             UPAUG1      17
          C SHIFT STORAGE                                                               FTPS3       38
                DO 460 I=1,NPTSS                                                        FTPS3       39
            460 UM1(I)=USI(I)                                                           FTPS3       40
          C                                                                             FTPS3       46
          C  FORM TRIDIAGONAL MATRIX                                                    FTPS3       47
          C                                                                             FTPS3       48
                DO 511 I=2,NPTSS                                                        FTPS3       49
          C                                                                             FTPS3       57
                BBB=RCX(I)/DTT+RP(I)+RM(I)                                              FTPS3       58
                AAAA(I)=RM(I)/BBB                                                       FTPS3       59
                CCCC(I)=RP(I)/BBB                                                       FTPS3       60
                DDDD(I)=RCX(I)*UM1(I)/DTT/BBB                                           UPSEP12      1
            511 CONTINUE                                                                FTPS3       72
          C  TRIDIAGONAL SOLUTION                                                       FTPS3       73
                DO 540 M=1,MIT                                                          FTPS3       74
          C  I=1                                                                        FTPS3       79
                BBB=RCX(1)/DTT+RP(1)+RM(1)+4.*E*SIG*(USI(1)+460.)**3.+                  UPAUG1      18
               @HBAR*HREF                                                               UPAUG1      19
                AAAA(1)=RM(1)/BBB                                                       UPAUG1      20
                CCCC(1)=RP(1)/BBB                                                       UPAUG1      21
                DDDD(1)=(RCX(1)*UM1(1)/DTT+E*SIG*(3*USI(1)**4+TRAD**4)+HBAR*HREF*       UPAUG1      22
               &TAW1)/BBB                                                               UPAUG1      23
                 G(1)=DDDD(1)                                                           UPAUG1      24
                 W(1)=-CCCC(1)                                                          UPAUG1      25
                DO 520 I=2,NPTSS                                                        FTPS3      114
            510 W(I)=-CCCC(I)/(1.+AAAA(I)*W(I-1))                                       FTPS3      115
            520 G(I)=(DDDD(I)+AAAA(I)*G(I-1))/(1.+AAAA(I)*W(I-1))                       FTPS3      116
                UNEW =G(NPTSS)                                                          FTPS3      118
                UERMX=ABS(UNEW-USI(NPTSS))                                              FTPS3      119
                USI(NPTSS)=UNEW                                                         FTPS3      120
                DO 530 L=2,NPTSS                                                        FTPS3      121
                I=NPTSS-L+1                                                             FTPS3      122
                UNEW=G(I)-W(I)*USI(I+1)                                                 FTPS3      123
                UERR=ABS(UNEW-USI(I))                                                   FTPS3      124
                UERMX=AMAX1(UERMX,UERR)                                                 FTPS3      125
                USI(I)=UNEW                                                             FTPS3      126
            530 CONTINUE                                                                FTPS3      127
                IF(UERMX.LT.ERALOW.AND.M.GE.5)GO TO 550                                 FTPS3      128
            540 CONTINUE                                                                FTPS3      129
```

```
1              RCX(I)=RHOG*PHIC*ZP*DX(1)*.5                          UPAUG1      8
1              RP(I)=PHIK*Z/DX(1)                                    UPAUG1      9
1              RM(I)=0.                                              FMAKEA2   181
1              A(1,1)=-RP(I)/RCX(I)                                  FMAKEA2   182
1              A(1,2)=RP(I)/RCX(I)                                   FMAKEA2   183
1              ALIN1=-(4.*E*SIG*(USI(1)+460.)**3.+HBAR*HREF)/RCX(I)  OCT30      14
1              END IF                                                FMAKEA2   185
           511 CONTINUE                                              FMAKEA2   186
               A(1,1) = A(1,1)+ALIN1                                 FMAKEA2   187
         C                                                           FMAKEA2   193
         C  SYSTEM MATRIX COMPLETED                                  FMAKEA2   194
         C                                                           FMAKEA2   195
               RETURN                                                FMAKEA2   196
               END                                                   FMAKEA2   197
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
             SUBROUTINE MAKEA                                                    FMAKEA2      2
             COMMON /CPLCL/IPT,PFRAC                                             FMAKEA2      3
             COMMON/COMTUN/T,TAW1,ALPHA,H,V,RHO,P,TEMP,C,TRAD,RHOG,              OCT10        3
            &TO,TSINK,XFT,DEL,PDEL                                               COMTUN       3
             COMMON /CHEAT/Q,TS,QREF,TW,M1,RENS,HBAR,HREF                        UPAUG16     40
             COMMON /ICTPS2/TINIT(1),ERALOW,E                                    UPAUG16     41
             COMMON/COSP/NPTSS,USI(6),PHI(6,6),NPT,PC(6,6),RR,                   UPAUG16      6
            &QD(6,6),QDT(6,6),QUE(6,6),A(6,6),RCX(6),RP(6),RM(6)                 UPOCT09      5
             COMMON/CPC/NPTPC                                                    FMAKEA2     13
             LOGICAL FAUTO                                                       CPARAM       2
             DIMENSION FAUTO(7)                                                  OCT10        2
             DIMENSION QP(5)                                                     UPAUG16      1
             COMMON/CPARAM/HO,HALF(2),PHIC,PHIK,ZP,Z,ALPH(2),KA,S(5),            UPAUG16      2
            &CIF(5,5),KAF,IFX(5),ACC(5),IFXSUM,NPAR,DALPH(2)                     UPAUG16      3
             EQUIVALENCE (HO,QP(1))                                              UPAUG16      4
             COMMON/CKF/K(6),S1(6),J1(6,6),TC(2),NODES(2),KFOPT                  UPAUG16      8
             REAL K,J1                                                           FDKF         3
             COMMON /CDX/DX(1)                                                   UPAUG15      2
             REAL M1                                                             FMAKEA2     18
             COMMON /CTCMMT/NTCT                                                 UPAUG16     42
             DATA SIG/4.761E-13/                                                 UPAUG16     43
             DATA E/.3/                                                          UPAUG16     44
             DATA HREF/1./                                                       UPAUG20      4
             DATA RHOG/17.10603937/                                              OCT30       11
             DATA ZP/3.233477/                                                   OCT30       12
             DATA Z/3.054E-3/                                                    OCT30       13
       C  SET UP LINEARIZED SYSTEM MATRIX, A                                      FMAKEA2     44
       C                                                                         FMAKEA2     45
             CALL ZERO (A(1,1),NPTSS,NPTSS)                                      FMAKEA2     46
             DO 511 I=1,NPTSS                                                    FMAKEA2     47
       C  CURRENT PASS TEMPERATURES                                              FMAKEA2     48
       C                                                                         FMAKEA2     57
       C  FORM MATRIX                                                            FMAKEA2     58
       C                                                                         FMAKEA2     59
       C                                                                         FMAKEA2     94
       C  I=NPTS                                                                 FMAKEA2     95
             IF (I.EQ.NPTSS) THEN                                                UPAUG1       1
1            RCX(I)=RHOG*PHIC*ZP*DX(1)*.5                                        UPAUG1       2
1            RM(I)=PHIK*Z/DX(1)                                                  UPAUG1       3
1            RP(I)=0.                                                            FMAKEA2    104
1            A(I,I-1)=RM(I)/RCX(I)                                               FMAKEA2    105
1            A(I,I)=-RM(I)/RCX(I)                                                FMAKEA2    106
1      C                                                                         FMAKEA2    125
1      C  BLOCK B  INTERIOR POINTS                                               FMAKEA2    126
1      C                                                                         FMAKEA2    128
1             ELSE IF((I.GT.1).AND.(I.LT.NPTSS))THEN                             UPAUG1       4
1            RCX(I)=RHOG*PHIC*ZP*DX(1)                                           UPAUG1       5
1            RP(I)=PHIK*Z/DX(1)                                                  UPAUG1       6
1            RM(I)=PHIK*Z/DX(1)                                                  UPAUG1       7
1            A(I,I-1)=RM(I)/RCX(I)                                               FMAKEA2    170
1            A(I,I+1)=RP(I)/RCX(I)                                               FMAKEA2    171
1            A(I,I)=-A(I,I-1)-A(I,I+1)                                           FMAKEA2    172
1      C                                                                         FMAKEA2    173
1      C SURFACE NODE I=1                                                        FMAKEA2    174
1            ELSE IF (I.EQ.1) THEN                                               FMAKEA2    175
```

```
      C  WHILE I.EQ.1 ADD BOUNDARY NOISE DUE TO HEAT TERMS          FQUE     54
            IF (I.EQ.1) THEN                                        FQUE     55
 1          QUE(I,J) = QER*QER*SDME*SQRT(SDBN**2+SDME**2)*R(I,J)    FQUE     56
 1          ELSE                                                    FQUE     57
 1          QUE(I,J) = QER*QER*SDME**2.*R(I,J)                      FQUE     58
 1          END IF                                                  FQUE     59
 1    C  END WHILE                                                  FQUE     60
 1    C                                                             FQUE     61
        221 QUE(J,I) = QUE(I,J)                                     FQUE     62
        222 CONTINUE                                                FQUE     63
      C                                                             FQUE     64
            RETURN                                                  FQUE     65
            END                                                     FQUE     66
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
      SUBROUTINE QUEMAT                                                FQUE        2
       COMMON /CONST/ XP1(13)                                          HAROLD     28
      LOGICAL IFICENT,IFPLOT,IFPRINT                                   UPOCT09    23
      COMMON /CFLAG/IFICENT,IFPLOT,IFPRINT                             UPOCT09    24
      COMMON/COMTUN/T,TAW1,ALPHA,H,V,RHO,P,TEMP,C,TRAD,RHOG,           OCT10       3
     &TO,TSINK,XFT,DEL,PDEL                                            COMTUN      3
      COMMON /CTCMMT/NTCT                                              UPAUG16    35
      COMMON /CDX/DX(1)                                                UPAUG16    36
      COMMON/COSP/NPTSS,USI(6),PHI(6,6),NPT,PC(6,6),RR,                UPAUG16     6
     &QD(6,6),QDT(6,6),QUE(6,6),A(6,6),RCX(6),RP(6),RM(6)              UPOCT09     5
      COMMON /ICTPS2/TINIT(1),ERALOW,E                                 UPAUG16    37
      COMMON/CSENS/SUSI(6,5),UM1(6)                                    UPAUG16     7
      COMMON/CKF/K(6),S1(6),J1(6,6),TC(2),NODES(2),KFOPT               UPAUG16     8
      REAL K,J1                                                        FDKF        3
      COMMON /CICSTAT/TR,SDIC,SDME,SDMEA,SDBN                          FQUE       15
      COMMON /CHEAT/Q,TS,QREF,TW,M1,RENS,HBAR,HREF                     UPAUG16    38
      REAL M1                                                          FQUE       17
      DIMENSION R(6,6)                                                 UPOCT09     7
      EQUIVALENCE (QD(1,1),R(1,1))                                     FQUE       19
C                                                                      FQUE       20
      RR = SDMEA                                                       FQUE       21
C                                                                      FQUE       22
C  MATRIX OF SPACIAL CORRELATIONS, R                                   FQUE       23
C  SPACIAL CORRELATIONS IN RSI MUST ALLOW FOR VARIABLE NODE STRUCTURE  FQUE       24
      DO 200 I=1,NPTSS                                                 FQUE       25
      SUMR=0.                                                          FQUE       26
      DO 200 J=I,NPTSS                                                 FQUE       27
      XP=SUMR/TR                                                       OCT30       9
      XP=ABS(XP)                                                       FQUE       29
      IF(I .EQ. 1)XP1(J)=XP*TR                                         HAROLD     29
      IF(XP.GT.100.)XP=100.                                            FQUE       30
      R(I,J)=EXP(-XP)                                                  FQUE       31
      IF(ABS(RP(J)).LT.1.E-8)GO TO 200                                 FQUE       32
      SUMR=SUMR+RCX(J)/RP(J)                                           FQUE       33
  200 R(J,I)=R(I,J)                                                    FQUE       34
      PRINT*,'XP1= ',(XP1(I),I=1,NPTSS)                                HAROLD     30
C                                                                      FQUE       35
C  MODEL ERROR MATRIX, QUE                                             FQUE       36
C                                                                      FQUE       37
      RC=SQRT(ABS((RCX(1)+RCX(2))*.5))                                 FQUE       38
      QER=(HBAR*HREF*(TAW1-USI(1))/(RM(1)+RP(1))/2)**2                 OCT10      10
      DO 221 I = 1,NPTSS                                               FQUE       40
C                                                                      FQUE       41
C  WHILE I.EQ.1 ADD BOUNDARY NOISE DUE TO HEAT TERMS                   FQUE       42
      IF (I.EQ.1) THEN                                                 FQUE       43
      QUE(I,I)=((SDBN*QER)**2+(SDME*QER)**2)/2                         OCT30      10
      ELSE                                                             FQUE       45
      QUE(I,I) = (SDME*QER)**2.                                        FQUE       46
      END IF                                                           FQUE       47
C  END WHILE                                                           FQUE       48
C                                                                      FQUE       49
      IF (I.EQ.NPTSS) GO TO 222                                        FQUE       50
      IP1 = I+1                                                        FQUE       51
      DO 221 J = IP1,NPTSS                                             FQUE       52
C                                                                      FQUE       53
```

```
      C                                                                  FINPUT     129
      C  READ CALCOMP PLOT SPECIFICATIONS                                 FINPUT     130
      C                                                                  FINPUT     131
            READ(5,1000)IPLOT                                            FINPUT     132
            READ(5,2000)TSCALE,TMIN,TAXL                                 FINPUT     133
            READ(5,2000)YSCALE,YMIN,YAXL                                 FINPUT     134
            READ(5,2000)ASCALE,AMIN,AAXL                                 FINPUT     135
       1000 FORMAT(10X,I2,7(3X,I2))                                      FINPUT     140
      C                                                                  FINPUT     142
      C  READ TIMES                                                      FINPUT     143
      C                                                                  FINPUT     144
      C START-STOP TIMES / PRINT TIME STEP                               FINPUT     145
            READ(5,2000)TSTART,TSTOP,DTPENT                              FINPUT     146
      C  DATA FOR I.C. SMOOTHER                                          CHUCK       34
            READ(5,100)SMIC,TSMTH                                        UPSEP24      3
       100  FORMAT(6X,L1,8X,F10.5)                                       CHUCK       36
      C # OF ITERATIONS/FIX=0 FIXES  PARAMETER/AUTO FLAG FOR REFERENCE   FINPUT     147
            READ(5,2010)NRPITER,IFX,FAUTO,KFOPT                          FINPUT     148
      C NEWTON-RAPHSON ACCELERATION PARAMETERS(0)ACC)2)                  FINPUT     149
            READ(5,2015)ACC                                              FINPUT     150
      C HEATING MODEL INITIAL PARAMETERS                                 FINPUT     151
            READ(5,2020)(QP(II),II=1,NPAR)                               UPSEP24      4
      C INITIAL REFERENCE VALUES FOR HEATING MODEL                       FINPUT     153
            READ(5,2030)ALPH                                             UPSEP24      5
       2010 FORMAT(5X,I2,3X,5I1,5X,7L1,8X,I1)                            OCT10        8
       2015 FORMAT(5X,8F5.2/5X,8F5.2)                                    FINPUT     156
       2020 FORMAT(10X,8F8.4/26X,5F8.4)                                  FINPUT     157
       2030 FORMAT(10X,7F8.4)                                            FINPUT     158
            RETURN                                                       FINPUT     159
            END                                                          FINPUT     160
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
      SUBROUTINE IC                                                     FIC        2
      LOGICAL IFICENT,IFPLOT,IFPRINT                                    UPOCT09   21
      COMMON /CFLAG/IFICENT,IFPLOT,IFPRINT                              UPOCT09   22
      COMMON /CTCMMT/NTCT                                               UPAUG16   31
      COMMON /CDX/DX(1)                                                 UPAUG16   32
      COMMON/COSP/NPTSS,USI(6),PHI(6,6),NPT,PC(6,6),RR,                 UPAUG16    6
     &QD(6,6),QDT(6,6),QUE(6,6),A(6,6),RCX(6),RP(6),RM(6)              UPOCT09    5
      COMMON /ICTPS2/TINIT(1),ERALOW,E                                  UPAUG16   33
      COMMON/CSENS/SUSI(6,5),UM1(6)                                     UPAUG16    7
      COMMON/CKF/K(6),S1(6),J1(6,6),TC(2),NODES(2),KFOPT                UPAUG16    8
      REAL K,J1                                                         FDKF       3
      COMMON /CICSTAT/TR,SDIC,SDME,SDMEA,SDBN                           FIC       15
      COMMON /CHEAT/Q,TS,QREF,TW,M1,RENS,HBAR,HREF                      UPAUG16   34
      REAL M1                                                           FIC       17
      DIMENSION R(6,6)                                                  UPOCT09    7
      EQUIVALENCE (QD(1,1),R(1,1))                                      FIC       19
C                                                                       FIC       20
C  INITIAL SENSITIVITIES                                                FIC       21
C                                                                       FIC       22
      RR = SDMEA                                                        FIC       23
C                                                                       FIC       24
C  MATRIX OF SPACIAL CORRELATIONS, R                                    FIC       25
C                                                                       FIC       26
C  SPACIAL CORRELATIONS IN RSI MUST ALLOW FOR VARIABLE NODE STRUCTURE   FIC       27
      DO 200 I=1,NPTSS                                                  FIC       28
      SUMR=0.                                                           FIC       29
      DO 200 J=I,NPTSS                                                  FIC       30
      XP=SUMR/TR                                                        OCT30      8
      XP=ABS(XP)                                                        FIC       32
      IF(XP.GT.100.)XP=100.                                             FIC       33
      R(I,J)=EXP(-XP)                                                   FIC       34
      IF(ABS(RP(J)).LT.1.E-8)GO TO 200                                  FIC       35
      SUMR=SUMR+RCX(J)/RP(J)                                            FIC       36
  200 R(J,I)=R(I,J)                                                     FIC       37
C                                                                       FIC       38
C  COVARIANCE MATRIX OF TEMP ICS, PC                                    FIC       39
C                                                                       FIC       40
      DO 210 I = 1,NPTSS                                                FIC       41
      PC(I,I) = (SDIC*USI(I))**2.                                       FIC       42
      IF (I.EQ.NPTSS) GO TO 211                                         FIC       43
      IP1 = I+1                                                         FIC       44
      DO 210 J = IP1,NPTSS                                              FIC       45
      PC(I,J) = USI(I)*USI(J)*SDIC**2.*R(I,J)                           FIC       46
  210 PC(J,I) = PC(I,J)                                                 FIC       47
  211 CONTINUE                                                          FIC       48
      RETURN                                                            FIC       49
      END                                                               FIC       50
```

```
      C                                                                        FINPUT     129
      C  READ CALCOMP PLOT SPECIFICATIONS                                      FINPUT     130
      C                                                                        FINPUT     131
            READ(5,1000)IPLOT                                                  FINPUT     132
            READ(5,2000)TSCALE,TMIN,TAXL                                       FINPUT     133
            READ(5,2000)YSCALE,YMIN,YAXL                                       FINPUT     134
            READ(5,2000)ASCALE,AMIN,AAXL                                       FINPUT     135
       1000 FORMAT(10X,I2,7(3X,I2))                                            FINPUT     140
      C                                                                        FINPUT     142
      C  READ TIMES                                                            FINPUT     143
      C                                                                        FINPUT     144
      C START-STOP TIMES / PRINT TIME STEP                                     FINPUT     145
            READ(5,2000)TSTART,TSTOP,DTPENT                                    FINPUT     146
      C  DATA FOR I.C. SMOOTHER                                                CHUCK       34
            READ(5,100)SMIC,TSMTH                                              UPSEP24      3
       100  FORMAT(6X,L1,8X,F10.5)                                             CHUCK       36
      C # OF ITERATIONS/FIX=0 FIXES  PARAMETER/AUTO FLAG FOR REFERENCE         FINPUT     147
            READ(5,2010)NRPITER,IFX,FAUTO,KFOPT                                FINPUT     148
      C NEWTON-RAPHSON ACCELERATION PARAMETERS(0)ACC)2)                        FINPUT     149
            READ(5,2015)ACC                                                    FINPUT     150
      C HEATING MODEL INITIAL PARAMETERS                                       FINPUT     151
            READ(5,2020)(QP(II),II=1,NPAR)                                     UPSEP24      4
      C INITIAL REFERENCE VALUES FOR HEATING MODEL                             FINPUT     153
            READ(5,2030)ALPH                                                   UPSEP24      5
       2010 FORMAT(5X,I2,3X,5I1,5X,7L1,8X,I1)                                  OCT10        8
       2015 FORMAT(5X,8F5.2/5X,8F5.2)                                          FINPUT     156
       2020 FORMAT(10X,8F8.4/26X,5F8.4)                                        FINPUT     157
       2030 FORMAT(10X,7F8.4)                                                  FINPUT     158
            RETURN                                                             FINPUT     159
            END                                                                FINPUT     160
```

```
SUBROUTINE INPUT      74/855  OPT=0,ROUND= A/ S/ M/-D,-DS     FTN 5.1+587         84/11/19. 13.14.29        PAGE     1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

            SUBROUTINE INPUT                                                  FINPUT      2
            LOGICAL IFICENT,IFPLOT,IFPRINT                                    FINPUT      3
            CHARACTER *30 VEH,FLTDT,TMANV,CTPT                                FINPUT      5
            COMMON /CFLAG/IFICENT,IFPLOT,IFPRINT                              UPOCT09    20
            COMMON/CTCMMT/NTCT                                                UPAUG16    26
            COMMON/CTIME/TSTART,TSTOP,DTPENT,NRPITER,ITPRAM                   FINPUT      9
            COMMON/ICTPS2/TINIT(1),ERALOW,E                                   UPAUG16    27
            COMMON/CDX/DX(1)                                                  UPAUG16    28
            LOGICAL FAUTO                                                     CPARAM      2
            DIMENSION FAUTO(7)                                                OCT10       2
            DIMENSION QP(5)                                                   UPAUG16     1
            COMMON/CPARAM/HO,HALF(2),PHIC,PHIK,ZP,Z,ALPH(2),KA,S(5),          UPAUG16     2
           &CIF(5,5),KAF,IFX(5),ACC(5),IFXSUM,NPAR,DALPH(2)                   UPAUG16     3
            EQUIVALENCE (HO,QP(1))                                            UPAUG16     4
            COMMON/CKF/K(6),S1(6),J1(6,6),TC(2),NODES(2),KFOPT                UPAUG16     8
            REAL K,J1                                                         FDKF        3
            COMMON /CSMTH/ UICSM(6),PICSM(6,6),UAP(6),PAP(6,6),               UPOCT09     1
           &SMIC,TSMTH,W(6,6)                                                 UPOCT09     2
            LOGICAL SMIC                                                      FCSMTH      4
            COMMON /CCOM/VEH,FLTDT,TMANV,CTPT                                 FINPUT     16
            COMMON /CICSTAT/TR,SDIC,SDME,SDMEA,SDBN                           FINPUT     17
            COMMON /CFPLOT/IPLOT,TSCALE,TMIN,TAXL,YSCALE,YMIN,YAXL,ASCALE,AMIN FINPUT    18
           *,AAXL                                                             FINPUT     19
      C                                                                       FINPUT     22
      C  READ VEHICLE/MANEUVER UNIQUES                                        FINPUT     23
      C                                                                       FINPUT     24
            READ(5,3000)VEH                                                   FINPUT     25
       3000 FORMAT(A30)                                                       FINPUT     26
            READ(5,3000)FLTDT                                                 FINPUT     27
            READ(5,3000)TMANV                                                 FINPUT     28
            READ(5,3000)CTPT                                                  FINPUT     29
      C                                                                       FINPUT     30
      C                                                                       FINPUT     49
            READ(5,1000)NTCT                                                  UPAUG24     2
      C                                                                       FINPUT     64
      C  IFICENT = T  ICS FROM DISK       / F  CONSTANT ICS FROM CARDS        FINPUT     65
      C  IFPLOT  = T  CREATE CALCOMP      / F  NO PLOT FILE                   FINPUT     66
      C  IFPRINT = T  PRINT TIME SERIES   / F  NO TEMP TIME SERIES OUTPUT     FINPUT     67
      C                                                                       FINPUT     68
            READ(5,4000)IFICENT                                               FINPUT     69
       4000 FORMAT(8(9X,L1))                                                  FINPUT     70
            READ(5,4000)IFPLOT                                                FINPUT     71
            READ(5,4000)IFPRINT                                               FINPUT     72
       2000 FORMAT(10X,F10.5,3(5X,F10.5)/10X,F10.5,3(5X,F10.5))               FINPUT     73
      C                                                                       FINPUT     74
      C  READ IN INITIAL TEMPERATURES  (USED IF IFICENT = F)                  FINPUT     75
      C                                                                       FINPUT     76
            READ(5,2000)TINIT                                                 FINPUT     77
      C                                                                       FINPUT     78
      C                                                                       FINPUT    122
      C  INITIAL COVARIANCE AND ERROR STATISTICS                              FINPUT    123
      C                                                                       FINPUT    124
            READ(5,2001)TR,SDIC,SDME,SDMEA                                    FINPUT    125
            READ(5,2001)SDBN                                                  FINPUT    126
       2001 FORMAT(10X,F10.5,3(5X,F10.5))                                     FINPUT    127
```

```
      PRINT *,'T,USI=',TLEST,(USI(II),II=1,NPTSS)                    FHEATEST2 450
      REWIND 3                                                       FHEATEST2 451
         TSTART=TSTOP                                                FHEATEST2 452
         GO TO 4                                                     FHEATEST2 453
         END IF                                                      FHEATEST2 454
      REWIND 3                                                       FHEATEST2 455
      WRITE(3) USI,PC,SUSI                                           FHEATEST2 456
      PRINT *,'T,USI=',TLEST,(USI(II),II=1,NPTSS)                    FHEATEST2 457
  998 IF(IFPLOT) CALL FPLOT                                           CHUCK      32
C                                                                    FHEATEST2 459
C  PLOT CALCOMP FILE FROM TAPE12                                     FHEATEST2 460
C                                                                    FHEATEST2 461
      STOP                                                           FHEATEST2 462
  999 STOP 'END-OF-FILE ENCOUNTERED ON INPUT TAPE IN HEATEST'         FHEATEST2 463
      END                                                            FHEATEST2 464
```

```
1                IF(NLAB.EQ.0)GO TO 120                                           FHEATEST2 342
1             24 CONTINUE                                                         FHEATEST2 343
1                   GO TO 100                                                     FHEATEST2 344
1                END IF                                                           FHEATEST2 345
1          C                                                                      FHEATEST2 346
1          C PROPAGATION TO TSTOP                                                 FHEATEST2 347
1          C                                                                      FHEATEST2 348
            120  DELT=TSTOP-TLEST                                                 FHEATEST2 349
                 CALL TPS3(DELT)                                                  FHEATEST2 350
                 TLEST=TSTOP                                                      FHEATEST2 351
           C  WRITE SMOOTHED I.CS TO TAPE3                                        CHUCK      23
                 IF(SMIC)THEN                                                     CHUCK      24
1                  REWIND (3)                                                     CHUCK      27
1                  WRITE(3) UICSM,PICSM,SUSI                                      CHUCK      28
1                  REWIND (3)                                                     CHUCK      29
1                 IIC=1                                                           HAROLD     26
1                ENDIF                                                            CHUCK      30
1          C                                                                      FHEATEST2 352
1          C  EXIT TEMPERATURE/STATE ESTIMATION LOOP                              FHEATEST2 353
1          C                                                                      FHEATEST2 354
                 IF(ITPRAM.LE.NRPITER.OR.NRPITER.EQ.0) THEN                       FHEATEST2 355
           C                                                                      FHEATEST2 356
           C  UPDATE PARAMETER ESTIMATES - LIST RESULTS                           FHEATEST2 357
           C                                                                      FHEATEST2 358
1                CALL PAREST                                                      FHEATEST2 359
1                IT=1                                                             FHEATEST2 360
1                DO 197 II=1,NPAR                                                 FHEATEST2 361
1                IF(IFX(II).EQ.0)THEN                                             FHEATEST2 362
2                CI(II)=0.                                                        FHEATEST2 363
2                GO TO 197                                                        FHEATEST2 364
2                END  IF                                                          FHEATEST2 365
1                CI(II)=SQRT(ABS(CIF(IT,IT)))                                     FHEATEST2 366
1                IT=IT+1                                                          FHEATEST2 367
1           197  CONTINUE                                                         FHEATEST2 368
1                IF(ITPRAM.EQ.1)WRITE(6,3079)                                     FHEATEST2 371
1          3079 FORMAT(//1X,'ITER',T18,'HALPH1',T30,'HALPH2',T42,'PHIC',          OCT30       6
1              &T54,'PHIK',T66,'ZP',T76,'Z',T88,'ALPHA1',T100,                    OCT30       7
1              &'ALPHA2',T112,'ALPHA3'/T6,'QBETA',T18,'QLOGRE',T30,'QDELE',T42    FHEATEST2 374
1              &,'QDELBF',T54,'QMACH',T66,'PHIKB'/1X,'(CRAMER-RAO BOUND)')        FHEATEST2 375
1                WRITE(6,3080)ITPRAM,(QP(I),I=1,10),(CI(I),I=1,10),               FHEATEST2 376
1              &(QP(I),I=11,16),(CI(I),I=11,16)                                   FHEATEST2 377
1          3080 FORMAT(1X,I2,2X,10(F8.5,4X)/5X,10(1X,'(',E8.2,')',1X)/            FHEATEST2 378
1              &/5X,6(F8.5,4X)/5X,6(1X,'(',E8.2,')',1X))                          FHEATEST2 379
1                WRITE(6,3081)AVERROR                                             FHEATEST2 380
1          3081 FORMAT(1X,'AVERAGE ERROR=',E12.5/)                                FHEATEST2 381
1          C                                                                      FHEATEST2 382
1                END IF                                                           FHEATEST2 383
             198 CONTINUE                                                         FHEATEST2 384
                 IIC=1                                                            FHEATEST2 385
           C                                                                      FHEATEST2 397
           C                                                                      FHEATEST2 444
                 IF(SMIC) GO TO 998                                               CHUCK1      1
                 IF(TSTOP.LT.TSTOPF-1.E-6)THEN                                    FHEATEST2 445
           C  RESET INITIAL CONDITIONS WITH FOLLOWING DATA TO DISK                FHEATEST2 446
           C                                                                      FHEATEST2 447
1                REWIND 3                                                         FHEATEST2 448
1                WRITE(3)USI,PC,SUSI                                              FHEATEST2 449
```

```
2          C  SAVE A PRIORI VALUES                                                CHUCK       11
1              DO 27 I=1,NPTSS                                                    CHUCK       12
1               UAP(I)=USI(I)                                                     CHUCK       13
1              DO 27 J=1,NPTSS                                                    CHUCK       14
1          27  PAP(I,J)=PC(I,J)                                                   CHUCK       15
1          C                                                                      CHUCK       16
1              DO 25 I=1,NTCT                                                     FHEATEST2  312
1              PU(I)=USI(NODES(I))                                                FHEATEST2  313
1           25         PE(I)=SQRT(ABS(PC(NODES(I),NODES(I))))                     FHEATEST2  314
1              CALL KF                                                            FHEATEST2  315
1              E1 = SQRT(ABS(PC(1,1)))                                            FHEATEST2  316
1              DO 26 I=1,NTCT                                                     FHEATEST2  317
1              PUF(I)=USI(NODES(I))                                               FHEATEST2  318
1           26 PEF(I)=SQRT(ABS(PC(NODES(I),NODES(I))))                            FHEATEST2  319
1              WRITE(6,99)T,PU(1),PE(1),PUF(1),PEF(1),TC(1)                       OCT30        2
1           99 FORMAT(1X,F5.2,3X,2(3X,F9.5,3X,'(',E8.2,')'),5X,F9.5)              OCT30        3
1              WRITE(6,98)(USI(I),I=1,NPTSS)                                      OCT30        4
1           98 FORMAT(T11,6(F8.5,3X))                                             OCT30        5
1          C INITIALIZE SMOOTHER                                                  HAROLD      10
1              IF(ICOUNT .EQ. 0)THEN                                              HAROLD      11
2               DO 1000 I=1,NPTSS                                                 HAROLD      12
2                UICSM(I)=USI(I)                                                  HAROLD      13
2               DO 1000 J=1,NPTSS                                                 HAROLD      14
2                PICSM(I,J)=PC(I,J)                                               HAROLD      15
2           1000 W(I,J)=PC(I,J)                                                   HAROLD      16
2              ENDIF                                                              HAROLD      17
1               TSMTH=TLEST                                                       HAROLD      18
1          C  SMOOTH I.C.'S                                                       CHUCK       17
1              IF(SMIC)THEN                                                       CHUCK       18
2               IF(ICOUNT .NE. 0)THEN                                             HAROLD      19
3               CALL FPSM(TSTART)                                                 HAROLD      20
3              DO 510 IM=1,NPTSS                                                  HAROLD      21
3           510 VAR(IM)=SQRT(ABS(PICSM(IM,IM)))                                   HAROLD      22
3               ENDIF                                                             CHUCK       21
2              ENDIF                                                              CHUCK       22
2          C                                                                      FHEATEST2  320
2          C  WRITE APOSTERIORI STATE ESTIMATE TO TAPE10/TAPE12                   FHEATEST2  321
2          C                                                                      FHEATEST2  322
1              IF (ITPRAM.GT.NRPITER) THEN                                        FHEATEST2  323
2                IF(IFPRINT)WRITE(10)1,TLEST,USI(1),E1,EQUI,PU,PE,PUF,PEF,TC,     FHEATEST2  324
2            &      Q,QREF,ALPHA,BETA,RENS,DELE,DELBF,M1                          UPOCT09     16
2            &,M1                                                                 FHEATEST2  326
2              IF(IFPLOT)WRITE(12,3055)TLEST,USI(1),EQUI,PU(1),TC(1),             FHEATEST2  328
2            &ALPHA,BETA,RENS,DELE,DELBF,M1,QN                                    FHEATEST2  329
2                  TRITE=TLEST                                                    FHEATEST2  330
2              END IF                                                             FHEATEST2  331
1              DTP = 0.                                                           FHEATEST2  332
1          C                                                                      FHEATEST2  333
1               ICOUNT= 1                                                         HAROLD      25
1          C  KF UPDATE COMPLETE - SET UP FOR NEXT PROPAGATION INTERVAL           FHEATEST2  334
1          C                                                                      FHEATEST2  335
1          C UPDATE THERMAL PROPERTIES/A MATRIX BASED ON UPDATED STATES           FHEATEST2  336
1          C NOTE! PROPERTIES MAY NEED UPDATED MORE OFTEN IF TC SAMPLE RATE IS LOW FHEATEST2  337
1                 CALL MAKEA                                                      FHEATEST2  338
1          C        READ NEXT THERMOCOUPLE SAMPLE                                 FHEATEST2  339
1              DO 24 II=1,ITCSK                                                   FHEATEST2  340
1                READ(13,END=999)NLAB,TUPDT,(TC(I),I=1,NTCT)                      UPAUG24      1
```

```
                    DO 500 IM=1,NPTSS                                              HAROLD       4
                     UICSM(IM)=USI(IM)                                             HAROLD       5
              500    VAR(IM)=SQRT(ABS(PC(IM,IM)))                                  HAROLD       6
                    PRINT*,'UIC= ',(UICSM(IM),IM=1,NPTSS)                          HAROLD       7
                    PRINT*,'VAR= ',(VAR(IM),IM=1,NPTSS)                            HAROLD       8
                    ICOUNT=0                                                       HAROLD       9
            C PROPAGATION TO TRAJECTORY SAMPLE TIME/TIMES                          FHEATEST2  262
            C                                                                      FHEATEST2  263
              100  IF(T.LT.TUPDT.AND.T.LT.TSTOP)THEN                               FHEATEST2  264
 1                    DELT=T-TLEST                                                 FHEATEST2  265
 1                    CALL TPS3(DELT)                                              FHEATEST2  266
 1                    CALL SENS(DELT)                                              FHEATEST2  267
 1                 CALL TPSOSP2(DELT)                                              FHEATEST2  268
 1                    TLEST=T                                                      FHEATEST2  269
 1                 DTP=DTP+DELT                                                    FHEATEST2  270
 1          C                                                                      FHEATEST2  271
 1          C  WHEN DTP.GE.DTPENT THEN WRITE TEMP/STATE ESTIMATES TO TAPE10/TAPE12  FHEATEST2  272
 1          C                                                                      FHEATEST2  273
 1                 IF ((DTP.GE.DTPENT).AND.(ITPRAM.GT.NRPITER)) THEN               FHEATEST2  274
 2                      DTP = 0.                                                   FHEATEST2  275
 2                      IF(IFPRINT)WRITE(6)0,TLEST,USI(1),TC(1),HBAR,HREF,ALPHA,TO  UPOCT09     13
 2                      IF(IFPRINT)WRITE(10)0,TLEST,USI(1),EQUI,USI(NODES(1)),Q,   UPOCT09     14
 2                &QREF,ALPHA,BETA,RENS,DELE,DELBF,M1                              FHEATEST2  277
 2                     IF(IFPLOT)WRITE(12,3055)TLEST,USI(1),EQUI,USI(NODES(1)),TC(1),  FHEATEST2  279
 2                 &    ALPHA,BETA,RENS,DELE,DELBF,M1,QN                           FHEATEST2  280
              3055      FORMAT(6E13.7)                                             FHEATEST2  281
 2                      TRITE=TLEST                                                FHEATEST2  282
 2                 END IF                                                          FHEATEST2  283
 2          C                                                                      FHEATEST2  284
 2          C         READ NEXT TRAJECTORY SAMPLE                                  FHEATEST2  285
 1                 DO 22 II=1,ITRJSK                                               FHEATEST2  286
 1                    READ(4,END=999)NLAB,T,(FLT1(I),I=1,NLAB)                     FHEATEST2  287
 1                    IF(NLAB.EQ.0)T=TSTOP                                         FHEATEST2  288
 1             22  CONTINUE                                                        FHEATEST2  289
 1                 CALL HEATTUN                                                    UPAUG16     24
 1                    GO TO 100                                                    FHEATEST2  291
 1                 END IF                                                          FHEATEST2  292
 1          C          PROPAGATION TO THERMOCOUPLE SAMPLE TIME/TIMES               FHEATEST2  293
 1          C                                                                      FHEATEST2  294
                   IF(TUPDT.LE.TSTOP)THEN                                          FHEATEST2  295
 1                    DELT=TUPDT-TLEST                                             FHEATEST2  296
 1                    CALL TPS3(DELT)                                              FHEATEST2  297
 1                    CALL SENS(DELT)                                              FHEATEST2  298
 1                    CALL TPSOSP2(DELT)                                           FHEATEST2  299
 1                    TLEST=TUPDT                                                  FHEATEST2  300
 1                 DTP=DTP+DELT                                                    FHEATEST2  301
 1                 IF(TUPDT.GE.T)THEN                                              FHEATEST2  302
 2                 DO 23 II=1,ITRJSK                                               FHEATEST2  303
 2                    READ(4,END=999)NLAB,T,(FLT1(I),I=1,NLAB)                     FHEATEST2  304
 2                    IF(NLAB.EQ.0)T=TSTOP                                         FHEATEST2  305
 2              23 CONTINUE                                                        FHEATEST2  306
 2                 CALL HEATTUN                                                    UPAUG16     25
 2                 END IF                                                          FHEATEST2  308
 2          C                                                                      FHEATEST2  309
 2          C KALMAN UPDATES                                                       FHEATEST2  310
 2          C                                                                      FHEATEST2  311
 2          C                                                                      CHUCK       10
```

```
               CALL ZERO(CIF,NPAR,NPAR)                                  FHEATEST2 161
               IF(IIC.EQ.0)CALL ZERO(SUSI,NPTSS,NPAR)                    FHEATEST2 162
         C                                                               FHEATEST2 163
         C  READ C & B FILE LABEL ON TRAJ TAPE                           FHEATEST2 164
         C                                                               FHEATEST2 165
               READ(4)C,LIC,ITAIL,TITLE,ITEST,FLT,DFLT,DREQ,DCOM         FHEATEST2 166
            33 IF(EOF(4) .NE. 0.) CALL EXIT                              FHEATEST2 167
               IF (C .NE. CLAB) CALL EXIT                                FHEATEST2 168
               READ(4) LABEL,NRSECT,NREM,(REMARK(I),I=1,NREM),NLAB,      FHEATEST2 169
              *(PLAB1(I),PLAB2(I),NUM(I),I=1,NLAB)                       FHEATEST2 170
               IF (LABEL .NE. LABELX) CALL EXIT                          FHEATEST2 171
         C                                                               FHEATEST2 172
         C  READ C & B FILE LABEL ON THE T/C MEAS TAPE                   FHEATEST2 173
         C                                                               FHEATEST2 174
             7 READ(13)C,LIC,ITAIL,TITLE,ITEST,FLT,DFLT,DREQ,DCOM        FHEATEST2 175
            44 IF (EOF(13) .NE. 0.) CALL EXIT                            FHEATEST2 176
               IF(C .NE.CLAB) CALL EXIT                                  FHEATEST2 177
               READ(13)LABEL,NRSECT,NREM,(REMARK(I),I=1,NREM),NLAB,      FHEATEST2 178
              *(PLAB1(I),PLAB2(I),NUM(I),I=1,NLAB)                       FHEATEST2 179
               IF(LABEL .NE. LABELX) CALL EXIT                           FHEATEST2 180
         C                                                               FHEATEST2 181
         C  SET TEMPERATURE INITIAL CONDITIONS                           FHEATEST2 182
         C                                                               FHEATEST2 183
               IF(IIC.EQ.0)THEN                                          FHEATEST2 184
         C  INITIAL TEMPERATURES                                         FHEATEST2 185
         C                                                               FHEATEST2 186
               DO 403 I=1,NPTSS                                          FHEATEST2 187
           403 USI(I)=TINIT(1)                                           UPAUG16    21
               IF(IFICENT) THEN                                          FHEATEST2 189
               READ (3) USI                                              FHEATEST2 190
               END IF                                                    FHEATEST2 191
               END IF                                                    FHEATEST2 192
               IF(IIC.NE.0)THEN                                          FHEATEST2 193
                  REWIND 3                                               FHEATEST2 194
               READ(3)USI,PC,SUSI                                        FHEATEST2 195
                  REWIND 3                                               FHEATEST2 196
               END IF                                                    FHEATEST2 197
               EQUI=USI(1)                                               FHEATEST2 198
         C  INITIALIZE SMOOTHER                                          CHUCK       2
         C INITIALIZE PARAMETERS AT TSTART                               FHEATEST2 199
               DTP=0.                                                    FHEATEST2 200
               TLEST=TSTART                                              FHEATEST2 201
         C READ FIRST TRAJECTORY SAMPLE                                  FHEATEST2 202
          10   READ(4,END=999)NLAB,T,(FLT1(I),I=1,NLAB)                  FHEATEST2 203
               IF(NLAB.EQ.0)GO TO 999                                    FHEATEST2 204
               IF(T.LT.TSTART)GO TO 10                                   FHEATEST2 205
         C CALCULATE REFERENCE HEATING                                   FHEATEST2 206
               CALL HEATTUN                                              UPAUG16    22
         C READ FIRST THERMOCOUPLE SAMPLES AND LOCAL PRESSURE            FHEATEST2 208
          20   READ(13,END=999)NLAB,TUPDT,(TC(I),I=1,NTCT)               UPAUG16    23
               IF(NLAB.EQ.0)GO TO 999                                    FHEATEST2 210
               IF(TUPDT.LT.TSTART)GO TO 20                               FHEATEST2 211
         C INITIALIZE THERMAL PROPERTIES/A MATRIX                        FHEATEST2 212
               CALL MAKEA                                                FHEATEST2 213
                IF(IIC .EQ. 0)CALL IC                                    HAROLD      3
               CALL QUEMAT                                               FHEATEST2 215
         C                                                               FHEATEST2 261
```

```
      IF(UERMX.GT.ERALOW)WRITE(6,1000)ERALOW,N,T,UERMX,USI(1)          FTPS3     130
 1000 FORMAT(1X,15HMAX ERROR TEMP~,E12.6,2X,I2,2X,4(E12.6,1X))         FTPS3     131
 550  CONTINUE                                                          FTPS3     132
 570  RETURN                                                            FTPS3     133
      END                                                               FTPS3     134
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
      SUBROUTINE SENS(DTT)                                              UPAUG1     26
     COMMON/COMTUN/T,TAW1,ALPHA,H,V,RHO,P,TEMP,C,TRAD,RHOG,             OCT10       3
    &TD,TSINK,XFT,DEL,PDEL                                              COMTUN      3
     COMMON /CHEAT/Q,TS,QREF,TW,M1,RENS,HBAR,HREF                       UPAUG16    45
     COMMON/COSP/NPTSS,USI(6),PHI(6,6),NPT,PC(6,6),RR,                  UPAUG16     6
    &QD(6,6),QDT(6,6),QUE(6,6),A(6,6),RCX(6),RP(6),RM(6)                UPOCT09     5
     COMMON /ICTPS2/TINIT(1),ERALOW,E                                   UPAUG16    46
     COMMON /CDX/DX(1)                                                  UPAUG15     4
     COMMON/CSENS/SUSI(6,5),UM1(6)                                      UPAUG16     7
     LOGICAL FAUTO                                                      CPARAM      2
     DIMENSION FAUTO(7)                                                 OCT10       2
     DIMENSION QP(5)                                                    UPAUG16     1
     COMMON/CPARAM/HO,HALF(2),PHIC,PHIK,ZP,Z,ALPH(2),KA,S(5),           UPAUG16     2
    &CIF(5,5),KAF,IFX(5),ACC(5),IFXSUM,NPAR,DALPH(2)                    UPAUG16     3
     EQUIVALENCE (HO,QP(1))                                             UPAUG16     4
     REAL M1                                                            FSENS3     16
     DIMENSION AA(6),BB(6),CC(6),DD(6),AAA(6),CCC(6),DDD(6),W(6),G(6)   UPOCT09     9
     EQUIVALENCE (QD(1,1),AA(1)),(QD(1,2),BB(1)),(QD(1,3),CC(1)),       FSENS3     18
    &(QD(1,4),DD(1)),(QD(1,5),AAA(1))                                   FSENS3     19
     EQUIVALENCE (QDT(1,1),CCC(1)),(QDT(1,2),DDD(1)),(QDT(1,3),W(1)),   FSENS3     20
    &(QDT(1,4),G(1))                                                    FSENS3     21
     DATA SIG/4.761E-13/                                                UPAUG16    47
     DATA E/.3/                                                         UPAUG16    48
     DATA NPTS/40/                                                      FSENS3     23
C                                                                       FSENS3     25
      IF (DTT.EQ.0.0) GO TO 999                                         UPAUG1     28
C                                                                       FSENS3     28
C  BACKWARD-DIFFERENCE FORMULATION OF DIFF. EQS.                        FSENS3     29
C                                                                       FSENS3     30
C                                                                       FSENS3     35
C  I=1                                                                  FSENS3     36
     RCX(1)=RHOG*PHIC*ZP*DX(1)*.5                                       UPAUG1     29
     RP(1)=PHIK*Z/DX(1)                                                 UPAUG1     30
     RM(1)=0.                                                           FSENS3     43
C SET UP TRIDIAGONAL MATRIX (COMMON TERMS ONLY)                         FSENS3     49
     DO 520 I=1,NPTSS                                                   FSENS3     50
     BB(I)=RCX(I)/DTT+RM(I)+RP(I)                                       FSENS3     51
     AA(I)=RM(I)                                                        FSENS3     52
     CC(I)=RP(I)                                                        FSENS3     53
     DD(I)=RCX(I)/DTT                                                   FSENS3     54
 520 CONTINUE                                                           FSENS3     55
    . BB(1)=BB(1)+4.*E*SIG*(USI(1)+460.)**3+  HBAR  *HREF               UPAUG1     31
C                                                                       FSENS3     66
C  I=1 SENSITIVITY FOR EACH PARAMETER                                   FSENS3     67
     DO 530 IP=1,NPAR                                                   FSENS3     68
     IF(IFX(IP).EQ.0)GO TO 530                                          FSENS3     69
     DO 531 I=1,NPTSS                                                   FSENS3     70
 531 DDD(I)=DD(I)*SUSI(I,IP)                                            FSENS3     71
C                                                                       FSENS3     72
C  SENSITIVITY FOR UNIT SURFACE CONDUCTANCE NAUT (THETA ONE)            UPAUG1     32
      IF(IP.EQ.1) DDD(1)=DDD(1)+HREF*(TAW1-USI(1))                      UPAUG1     33
C SENSITIVITIES FOR EACH HEATING MODEL PARAMETER                        FSENS3     82
     IF(IP.EQ.2) DDD(1)=DDD(1) +HREF*(TAW1-USI(1))*(ALPHA-ALPH(1))      UPAUG1     34
     IF(IP.EQ.3) DDD(1)=DDD(1) +HREF*(TAW1-USI(1))*(ALPHA-ALPH(2))      UPAUG1     35
C  SENSITIVITY FOR SPECIFIC HEAT FACTOR PHIC                            UPAUG1     36
```

```
             IF(IP.EQ.4) THEN                                              UPAUG1    37
1            DDD(1)=DDD(1)-RP(1)*(USI(2)-USI(1))-(HBAR*HREF*(TAW1-         OCT30     15
1           &USI(1))+E*SIG*((USI(1)+460.)**4-TRAD**4))/PHIC                OCT30     16
1            DO 533 I=2, NPTSS-1                                           UPAUG1    40
1     533    DDD(I)=DDD(I)-(RM(I)*(USI(I-1)-USI(I))+RP(I)*(USI(I)-USI(     OCT30     17
1           &I+1)))/PHIC                                                   UPAUG1    42
1            DDD(NPTSS)=DDD(NPTSS)+RM(NPTSS)*(USI(NPTSS-1)-USI(NPTSS))/PHIC UPAUG1   43
1            END IF                                                        UPAUG1    44
1     C  SENSITIVITY FOR CONDUCTIVITY FACTOR PHIK                          UPAUG1    45
             IF(IP.EQ.5) THEN                                              UPAUG1    46
1            DDD(1)=DDD(1)+RP(1)*(USI(2)-USI(1))/PHIK                      UPAUG1    47
1            DO 534 I=2,NPTSS-1                                            UPAUG1    48
1     534    DDD(I)=DDD(I)+(RM(I)*USI(I-1)-(RM(I)+RP(I))*USI(I)+RP(I)      UPAUG1    49
1           &*USI(I+1))/PHIK                                               UPAUG1    50
1            DDD(NPTSS)=DDD(NPTSS)+RM(NPTSS)*(USI(NPTSS-1)-USI(NPTSS))/PHIK UPAUG1   51
1            END IF                                                        UPAUG1    52
             DO 535 I=1,NPTSS                                              FSENS3    105
             AAA(I)=AA(I)/BB(I)                                            FSENS3    106
             CCC(I)=CC(I)/BB(I)                                            FSENS3    107
       535   DDD(I)=DDD(I)/BB(I)                                           FSENS3    108
      C                                                                    FSENS3    109
      C                                                                    FSENS3    110
      C TRIDIAGONAL SOLUTION                                               FSENS3    111
      C ELIMINATION STEP                                                   FSENS3    112
             G(1)=DDD(1)                                                   FSENS3    113
             W(1)=-CCC(1)                                                  FSENS3    114
             DO 536 I=2,NPTSS                                              FSENS3    115
             W(I)=-CCC(I)/(1.+AAA(I)*W(I-1))                               FSENS3    116
       536   G(I)=(DDD(I)+AAA(I)*G(I-1))/(1.+AAA(I)*W(I-1))                FSENS3    117
      C BACKWARD SUBSTITUTION                                              FSENS3    118
             SUSI(NPTSS,IP)=G(NPTSS)                                       FSENS3    119
             DO 537 L=2,NPTSS                                              FSENS3    120
             I=NPTSS-L+1                                                   FSENS3    121
             SUSI(I,IP)=G(I)-W(I)*SUSI(I+1,IP)                             FSENS3    122
       537   CONTINUE                                                      FSENS3    123
       530   CONTINUE                                                      FSENS3    124
       999   RETURN                                                        FSENS3    125
             END                                                           FSENS3    126
```

```
SUBROUTINE PAREST      74/855   OPT=0,ROUND= A/ S/ M/-D,-DS      FTN 5.1+587        84/11/19. 13.14.29         PAGE      1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

                SUBROUTINE PAREST                                          FPAREST     2
                LOGICAL FAUTO                                              CPARAM      2
                DIMENSION FAUTO(7)                                         OCT10       2
                DIMENSION QP(5)                                            UPAUG16     1
                COMMON/CPARAM/HO,HALF(2),PHIC,PHIK,ZP,Z,ALPH(2),KA,S(5),   UPAUG16     2
               &CIF(5,5),KAF,IFX(5),ACC(5),IFXSUM,NPAR,DALPH(2)            UPAUG16     3
                EQUIVALENCE (HO,QP(1))                                     UPAUG16     4
                COMMON/ICTPS2/TINIT(1),ERALOW,E                            UPSEP24     7
                COMMON/CDX/DX(1)                                           UPSEP24     8
                COMMON/CTIME/TSTART,TSTOP,DTPENT,NRPITER,ITPRAM            FPAREST     8
                COMMON/MAIN2/IMA2                                          FPAREST     9
                DIMENSION CIFI(6,6)                                        UPSEP24     9
                DATA KIN,KOUT/5,6/                                         FPAREST    12
                IMA2=16                                                    FPAREST    13
                NR=IFXSUM                                                  FPAREST    14
                IF(NR.EQ.1)THEN                                            FPAREST    15
 1              CIF(1,1)=1./CIF(1,1)                                       FPAREST    16
 1              GO TO 20                                                   FPAREST    17
 1              END IF                                                     FPAREST    18
 1        C                                                                FPAREST    19
 1        C INVERT CONDITIONAL INFORMATION MATRIX,CIF                      FPAREST    20
                CALL GMINV(NR,NR,CIF,CIFI,MR,1)                            FPAREST    21
                DO 15 IR=1,NR                                              FPAREST    22
                DO 15 IC=1,NR                                              FPAREST    23
           15   CIF(IR,IC)=CIFI(IR,IC)                                     FPAREST    24
          20     IT=1                                                      UPSEP24    10
                DO 29 IP=1,NPAR                                            FPAREST    28
                IF(IFX(IP).EQ.0)GO TO 29                                   FPAREST    29
                JT=1                                                       FPAREST    30
                DO 28 JP=1,NPAR                                            FPAREST    31
                IF(IFX(JP).EQ.0)GO TO 28                                   FPAREST    32
                QP(IP)=QP(IP)+CIF(IT,JT)*S(JT)                             OCT24       3
                JT=JT+1                                                    FPAREST    35
           28   CONTINUE                                                   FPAREST    36
                IT=IT+1                                                    FPAREST    37
           29   CONTINUE                                                   FPAREST    38
                RETURN                                                     FPAREST    41
                END                                                        FPAREST    42
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

             SUBROUTINE KF                                                    FKFSUM       2
             LOGICAL IFICENT,IFPLOT,IFPRINT                                   FKFSUM       3
             COMMON /CSMTH/ UICSM(6),PICSM(6,6),UAP(6),PAP(6,6),              OCT10       11
            &SMIC,TSMTH,W(6,6)                                                OCT10       12
             COMMON /CFLAG/IFICENT,IFPLOT,IFPRINT                             UPOCT09     25
             COMMON /CTCMMT/NTCT                                              UPSEP24     12
             COMMON/CPC/NPTPC                                                 FKFSUM       8
             LOGICAL FAUTO                                                    CPARAM       2
             DIMENSION FAUTO(7)                                               OCT10        2
             DIMENSION QP(5)                                                  UPAUG16      1
             COMMON/CPARAM/HO,HALF(2),PHIC,PHIK,ZP,Z,ALPH(2),KA,S(5),         UPAUG16      2
            &CIF(5,5),KAF,IFX(5),ACC(5),IFXSUM,NPAR,DALPH(2)                  UPAUG16      3
             EQUIVALENCE (HO,QP(1))                                           UPAUG16      4
             COMMON /ICTPS2/TINIT(1),ERALOW,E                                 UPAUG24      3
             COMMON /CDX/DX(1)                                                UPAUG24      4
             COMMON/CSENS/SUSI(6,5),UM1(6)                                    UPAUG16      7
             COMMON/COSP/NPTSS,USI(6),PHI(6,6),NPT,PC(6,6),RR,                UPAUG16      6
            &QD(6,6),QDT(6,6),QUE(6,6),A(6,6),RCX(6),RP(6),RM(6)              UPOCT09      5
             COMMON/CTIME/TSTART,TSTOP,DTPENT,NRPITER,ITPRAM                  FKFSUM      16
             COMMON/CPARAM2/AVERROR,EQUI,UMEAS                                UPOCT09     26
             COMMON/CKF/K(6),S1(6),J1(6,6),TC(2),NODES(2),KFOPT               UPAUG16      8
             REAL K,J1                                                        FDKF         3
             COMMON /MAIN1/INP,IPVT(6),WORK(6)                                OCT10        4
             INP=6                                                            OCT10        5
       C                                                                      FKFSUM      21
             SDMEA=RR                                                         FKFSUM      22
       C                                                                      FKFSUM      23
       C  THE FOLLOWING CONTROL CONSTRUCT SORTS KF UPDATE ITERATIONS          FKFSUM      24
       C  REQUIRED BY VECTOR UPDATES AS SPECIFIED IN THE INPUT DECK           FKFSUM      25
       C                                                                      FKFSUM      26
             DO 98 I=1,NPTSS                                                  FKFSUM      29
             DO 5 ITT=1,NTCT                                                  FKFSUM      30
             IF(I.EQ.NODES(ITT))THEN                                          FKFSUM      31
1              NODE=NODES(ITT)                                                FKFSUM      32
1              UMEAS=TC(ITT)                                                  FKFSUM      33
1              GO TO 10                                                       FKFSUM      34
1            END IF                                                           FKFSUM      35
        5    CONTINUE                                                         FKFSUM      36
             GO TO 98                                                         FKFSUM      37
        10    ERROR=UMEAS-UAP(NODE)                                           HAROLD      38
       C                                                                      FKFSUM      39
             IF(KAF.EQ.1) AVERROR=ERROR                                       FKFSUM      40
             AVERROR=((KAF-1)*AVERROR+ERROR)/KAF                              FKFSUM      41
             KAF=KAF+1                                                        FKFSUM      42
       C                                                                      FKFSUM      43
       C  SCORE RUNNING SUMS FOR JACOBIAN OF LIKELIHOOD FN. S,                FKFSUM      44
       C  AND CONDITIONAL INFORMATION MATRIX, CIF                             FKFSUM      45
       C                                                                      FKFSUM      46
             R=(SDMEA*UMEAS)**2.                                              FKFSUM      47
       C                                                                      FKFSUM      48
             DO 26 KO=1,NPAR                                                  FKFSUM      49
             S1(KO)=SUSI(NODE,KO)*IFX(KO)*ERROR/(PC(NODE,NODE)+R)             FKFSUM      50
             DO 25 L=1,NPAR                                                   FKFSUM      51
         25  J1(KO,L)=SUSI(NODE,KO)*IFX(KO)*IFX(L)*SUSI(NODE,L)/(PC(NODE,NODE)+ FKFSUM    52
            *R)                                                               FKFSUM      53
```

```
   26  CONTINUE                                                    FKFSUM     54
       IT=1                                                        FKFSUM     55
       DO 29 IP=1,NPAR                                             FKFSUM     56
       IF(IFX(IP).EQ.0)GO TO 29                                    FKFSUM     57
       S(IT)=S1(IP)+S(IT)                                          FKFSUM     58
       JT=1                                                        FKFSUM     59
       DO 28 JP=1,NPAR                                             FKFSUM     60
       IF(IFX(JP).EQ.0)GO TO 28                                    FKFSUM     61
       CIF(IT,JT)=J1(IP,JP)+CIF(IT,JT)                             FKFSUM     62
       JT=JT+1                                                     FKFSUM     63
  28   CONTINUE                                                    FKFSUM     64
       IT=IT+1                                                     FKFSUM     65
  29   CONTINUE                                                    FKFSUM     66
C                                                                  FKFSUM     67
C  COMPUTE KALMAN GAIN, K                                          FKFSUM     68
C                                                                  FKFSUM     69
       DO 30 IK=1,NPTSS                                            FKFSUM     70
  30    K(IK)=PAP(IK,NODE)/(PAP(NODE,NODE)+R)                      HAROLD     37
C IF KOPT=1 UPDATE                                                 FKFSUM     72
C IF KOPT=2 UPDATE EXCEPT ON LAST ITERATION(ITPRAM-NRPITER)        FKFSUM     73
C IF KOPT=3 DO NOT UPDATE                                          FKFSUM     74
C IF KOPT=4 UPDATE COVARIANCE AND SENSITIVITY ONLY                 FKFSUM     75
C IF KFOPT=5 ONLY UPDATE TEMP ON LAST ITERATION                    FKFSUM     76
C IF KFOPT=6 ONLY UPDATE ON LAST ITERATION                         FKFSUM     77
       GO TO(101,102,103,104,101,101)KFOPT                         FKFSUM     78
  102  IF(ITPRAM.GT.NRPITER)GO TO 103                              FKFSUM     79
C                                                                  FKFSUM     80
C  STATE UPDATE                                                    FKFSUM     81
C                                                                  FKFSUM     82
  101  IF(KFOPT.GE.6.AND.ITPRAM.LE.NRPITER)GO TO 103               FKFSUM     83
       IF(KFOPT.EQ.5.AND.ITPRAM.LE.NRPITER)GO TO 104               FKFSUM     84
       DO 40 IO=1,NPTSS                                            FKFSUM     85
  40    USI(IO)=USI(IO)+K(IO)*(UMEAS-UAP(NODE))                    HAROLD     38
C                                                                  FKFSUM     87
C  SENSITIVITY UPDATE                                              FKFSUM     88
C                                                                  FKFSUM     89
  104  CONTINUE                                                    FKFSUM     90
       DO 35 IP=1,NPAR                                             FKFSUM     91
       DO 35 L=1,NPTSS                                             FKFSUM     92
    35 SUSI(L,IP)=SUSI(L,IP)-K(L)*SUSI(NODE,IP)                    FKFSUM     93
C                                                                  FKFSUM     94
C  COVARIANCE UPDATE, PC  -  JOSEPH FORM                           FKFSUM     95
       NPTSS=NPTPC                                                 FKFSUM     96
C                                                                  FKFSUM     97
       CALL ZERO(QD(1,1),NPTSS,NPTSS)                              FKFSUM     98
       DO 50 IC=1,NPTSS                                            FKFSUM     99
       QD(IC,IC) = 1.0                                             FKFSUM    100
    50 QD(IC,NODE) = QD(IC,NODE)-K(IC)                             FKFSUM    101
       CALL MAT4(NPTSS,NPTSS,PC(1,1),QD(1,1),QDT(1,1))             FKFSUM    102
       CALL MAT4(NPTSS,1,R,K(1),QD(1,1))                           FKFSUM    103
       DO 55 IPC=1,NPTSS                                           FKFSUM    104
       DO 55 JPC=1,NPTSS                                           FKFSUM    105
    55 PC(IPC,JPC) = QDT(IPC,JPC)+QD(IPC,JPC)                      FKFSUM    106
       NPTSS=INP                                                   FKFSUM    107
    98 CONTINUE                                                    FKFSUM    108
  103  CONTINUE                                                    FKFSUM    109
       RETURN                                                      FKFSUM    110
```

```
113          END                                        FKFSUM      111
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

            SUBROUTINE FPSM(TSTART)                                          HAROLD     39
            COMMON /CONST/ XP1(13)                                           HAROLD     40
           COMMON /CTCMMT/NTCT                                               OCT10      13
           COMMON /CSMTH/ UICSM(6),PICSM(6,6),UAP(6),PAP(6,6),               UPOCT09     1
          &SMIC,TSMTH,W(6,6)                                                 UPOCT09     2
           LOGICAL SMIC                                                      FCSMTH      4
           COMMON/COSP/NPTSS,USI(6),PHI(6,6),NPT,PC(6,6),RR,                 UPAUG16     6
          &QD(6,6),QDT(6,6),QUE(6,6),A(6,6),RCX(6),RP(6),RM(6)               UPOCT09     5
           COMMON/CKF/K(6),S1(6),J1(6,6),TC(2),NODES(2),KFOPT                UPAUG16     8
           REAL K,J1                                                         FDKF        3
           DIMENSION RINV(8,8),HTR(6,8),SFP(6,6),GAIN(6),                    UPOCT09     3
          &WRK1(6,6),WRK2(6,6)                                               UPOCT09     4
           COMMON /MAIN1/INP,IPVT(6),WORK(6)                                 OCT10       4
           INP=6                                                             OCT10       5
C                                                                            FPSMIC      8
C  THIS ROUTINE IS A FIXED POINT SMOOTHER ALGORITHM                          FPSMIC      9
C                                                                            FPSMIC     10
           CALL ZERO(SFP,NPTSS,NPTSS)                                        HAROLD     42
           DO 10 I=1,NTCT                                                    HAROLD     43
 10         SFP(NODES(I),NODES(I))=1./(RR*TC(I))**2.                         HAROLD     44
C                                                                            FPSMIC     25
C  FORM I-SP AND FIND PHIT                                                   FPSMIC     26
           CALL MMUL(SFP,PC,NPTSS,NPTSS,NPTSS,WRK1)                          FPSMIC     27
           DO 30 I=1,NPTSS                                                   FPSMIC     28
           DO 30 J=1,NPTSS                                                   FPSMIC     29
             WRK1(I,J)=-WRK1(I,J)                                            FPSMIC     30
             IF(I .EQ. J)WRK1(I,J)=1.0+WRK1(I,J)                             FPSMIC     31
 30        WRK2(I,J)=PHI(J,I)                                                HAROLD     45
           DO 35 I=1,NPTSS                                                   HAROLD     46
            DO 35 J=1,NPTSS                                                  HAROLD     47
 35          PHI(I,J)=WRK2(I,J)                                              HAROLD     48
C                                                                            FPSMIC     33
C  FORM W=W*PHIT*(I-SP)                                                      FPSMIC     34
           CALL MMUL(PHI,WRK1,NPTSS,NPTSS,NPTSS,WRK2)                        CHUCK1      4
           CALL MMUL(W,WRK2,NPTSS,NPTSS,NPTSS,WRK1)                          FPSMIC     36
           DO 40 I=1,NPTSS                                                   FPSMIC     37
           DO 40 J=1,NPTSS                                                   FPSMIC     38
 40        W(I,J)=WRK1(I,J)                                                  FPSMIC     39
C                                                                            FPSMIC     40
C  SOLVE FOR COVARIANCE -- P=P-W(S*PAP*S + S)WTRAN                           FPSMIC     41
           CALL MMUL(PAP,SFP,NPTSS,NPTSS,NPTSS,WRK1)                         FPSMIC     42
           CALL MMUL(SFP,WRK1,NPTSS,NPTSS,NPTSS,WRK2)                        FPSMIC     43
           DO 50 I=1,NPTSS                                                   FPSMIC     44
           DO 50 J=1,NPTSS                                                   FPSMIC     45
 50         WRK1(I,J)=WRK2(I,J) +SFP(I,J)                                    FPSMIC     46
            CALL TRI(NPTSS,WRK1,W,PHI,WRK2,NPTSS)                            HAROLD     49
           DO 60 I=1,NPTSS                                                   FPSMIC     48
           DO 60 J=1,NPTSS                                                   FPSMIC     49
 60         PICSM(I,J)=PICSM(I,J) - WRK2(I,J)                                FPSMIC     50
C                                                                            FPSMIC     51
C  SOLVE FOR SMOOTHED STATE (SCALAR UPDATES)                                 FPSMIC     52
           DO 150 I=1,NPTSS                                                  FPSMIC     53
            DO 110 ITT=1,NTCT                                                FPSMIC     54
             IF(I .EQ. NODES(ITT))THEN                                       FPSMIC     55
1              NODE=NODES(ITT)                                               FPSMIC     56
```

```
1                   UMEAS=TC(ITT)                                        FPSMIC    57
1                   GO TO 120                                            .PSMIC    58
1                  ENDIF                                                 FPSMIC    59
            110   CONTINUE                                               FPSMIC    60
                 GO TO 150                                               FPSMIC    61
            120  R=(RR*UMEAS)**2.                                        FPSMIC    62
           C                                                             FPSMIC    63
           C  COMPUTE GAIN                                               FPSMIC    64
                 DO 121  IJ=1,NPTSS                                      FPSMIC    65
            121   GAIN(IJ)=W(IJ,NODE)/R                                  FPSMIC    66
           C                                                             FPSMIC    67
           C  UPDATE                                                     FPSMIC    68
                 DO 125 IO=1,NPTSS                                       FPSMIC    69
                  TCONST=TSTART+XP1(IO)                                  HAROLD    50
                  IF(TSMTH .GT. TCONST) GO TO 125                        HAROLD    51
                  UICSM(IO)=UICSM(IO)+GAIN(IO)*(UMEAS-UAP(NODE))         HAROLD    52
            125   CONTINUE                                               HAROLD    53
            150  CONTINUE                                                FPSMIC    71
                 RETURN                                                  FPSMIC    72
                 END                                                     FPSMIC    73
```

```
SUBROUTINE MULT        74/855  OPT=0,ROUND= A/ S/ M/-D,-DS      FTN 5.1+587        84/11/19. 13.14.29         PAGE      1
DO--LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE MULT(X,Y,N,Z,N2)                                       MULT      2
      DIMENSION X(N2,N2),Y(N2,N2),Z(N2,N2                               MULT      3
      DO 20  I=1,N                                                      MULT      4
      DO 20  J=1,N                                                      MULT      5
      Z(I,J)=0.                                                         MULT      6
      DO 20  K=1,N                                                      MULT      7
   20 Z(I,J)=Z(I,J)+X(I,K)*Y(K,J)                                       MULT      8
      RETURN                                                            MULT      9
      END                                                               MULT     10
```

DO LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5 I ANSI=0,L=OUTS,LO=S/-A.

```
      SUBROUTINE MEXP(N,SUB1,TIME,SUB2,Q,QT,N2)                     MEXP2    2
      DIMENSION SUB1(N2,N2),SUB2(N2,N2)                             MEXP2    3
      DIMENSION Q(N2,N2),QT(N2,N2)                                  MEXP2    4
C     MULTIPLY ELEMENTS OF SUB1 BY TIME                             MEXP2    5
      DO 102  I=1,N                                                 MEXP2    6
      DO 102  J=1,N                                                 MEXP2    7
      SUB1(I,J)=SUB1(I,J)*TIME                                      MEXP2    8
  102 SUB2(I,J)=SUB1(I,J)                                           MEXP2    9
C     GENERATE IDENTITY MATRIX FOR INPUT Q, FOR HQR                 MEXP2   10
      DO 30  I=1,N                                                  MEXP2   11
      DO 30  J=1,N                                                  MEXP2   12
      Q(I,J)=0.                                                     MEXP2   13
   30 IF(I .EQ. J) Q(I,J)=1.                                        MEXP2   14
      CALL HQR(N,SUB2,Q,IERR,N2)                                    MEXP2   15
C     MATRIX SUB2 HAS BEEN DESTOYED                                 MEXP2   16
C     Q IS NOW AN ORTHOGONAL TRANSFORMATION MATRIX                  MEXP2   17
      DO 40  I=1,N                                                  MEXP2   18
      DO 40  J=1,N                                                  MEXP2   19
   40 QT(I,J)=Q(J,I)                                                MEXP2   20
C     QT IS NOW THE TRANSPOSE,AND THE INVERSE, OF Q                 MEXP2   21
      CALL MULT(SUB1,Q,N,SUB2,N2)                                   MEXP2   22
      CALL MULT(QT,SUB2,N,SUB1,N2)                                  MEXP2   23
C     SUB1 NOW CONTAINS THE TRIANGULAR MATRIX QT*A*Q                MEXP2   24
      DO 50  I=1,N                                                  MEXP2   25
      DO 50  J=1,N                                                  MEXP2   26
   50 SUB2(I,J)=0.                                                  MEXP2   27
      CALL FUNCT(1,N,SUB1,SUB2,N2)                                  MEXP2   28
C     SUB2 NOW HOLDS EXP(A*TIME) IN TRIANGULAR FORM                 MEXP2   29
      CALL MULT(SUB2,QT,N,SUB1,N2)                                  MEXP2   30
      CALL MULT(Q,SUB1,N,SUB2,N2)                                   MEXP2   31
C     SUB2 NOW HOLDS EXP(A*TIME) IN ORIGINAL BASIS FORM             MEXP2   32
      RETURN                                                        MEXP2   33
      END                                                           MEXP2   34
```

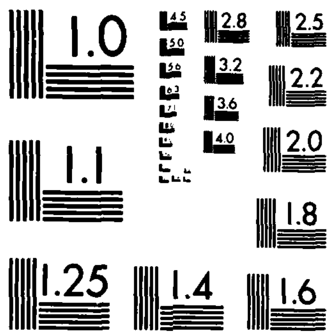
```
      SUBROUTINE FUNCT(R,S,T,F,MM)                                   FUNCT
      DIMENSION T(MM,MM),F(MM,MM)                                    FUNCT
      INTEGER R,S                                                    FUNCT
      REAL EXP                                                       FUNCT
      DO 10  I=R,S                                                   FUNCT
C THE IF-BLOCK GIVES 14-DIGIT ACCURACY WITHOUT UNDERFLOW              FUNCT
      IF( T(I,I) .LT. -43.) THEN                                     FUNCT
         F(I,I)=0.                                                   FUNCT
      ELSE                                                           FUNCT
        F(I,I)=EXP( T(I,I) )                                         FUNCT
      END IF                                                         FUNCT
   10 CONTINUE                                                       FUNCT
C   PROCESS THE KTH SUPERDIAGONAL                                    FUNCT
      N=S-R+1                                                        FUNCT
      NN=N-1                                                         FUNCT
C   NN = NUMBER OF SUPERDIAGONALS IN THE BLOCK                       FUNCT
      IF(NN .EQ. 0) RETURN                                           FUNCT
      DO 13  K=1,NN                                                  FUNCT
      LL=S-K                                                         FUNCT
         DO 12  I=R,LL                                               FUNCT
         DIFF=T(I,I)-T(I+K,I+                                        FUNCT
         IF(ABS(DIFF) .EQ. 0.0) GO TO 14                             FUNCT
         G=T(I,I+K)*(F(I,I)-F(I+K,I+K))                              FUNCT
         KK=K-1                                                      FUNCT
         IF(KK .EQ. 0) GO TO 12                                      FUNCT
            DO 11  M=1,KK                                            FUNCT
   11  G=G+(F(I,I+M)*T(I+M,I+K)-T(I,I+K-M)*F(I+K-M,I+K))             FUNCT
   12  F(I,I+K)=G/DIFF                                               FUNCT
   13 CONTINUE                                                       FUNCT
      RETURN                                                         FUNCT
   14 MM= -MM                                                        FUNCT
      RETURN                                                         FUNCT
      END                                                            FUNCT
```

```
      SUBROUTINE HQR(IGH,H,Z,IERR,N2)
      INTEGER I,J,K,L,M,N,EN,II,JJ,LL,MM,NA,NM,NN,N2,
     X        IGH,ITS,LOW,MP2,ENM2,IERR,MINO
      REAL H(N2,N2),Z(N2,N2)
      REAL P,Q,R,S,T,W,X,Y,RA,SA,VI,VR,ZZ,NORM
      REAL MACHEP,SQRT,ABS,SIGN,REAL,AIMAG
      LOGICAL NOTLAS
      COMPLEX Z3,CMPLX
C
C
C
C    MACHEP IS A PARAMETER THAT SPECIFIES PRECISION
      MACHEP=0.000000000001
      NM=IGH
      N=IGH
      LOW= 1
      IERR= 0
      NORM= 0.
      K= 1
C    COMPUTE MATRIX NORM
      DO 50  I = 1,N
         DO 40  J = K,N
   40    NORM = NORM + ABS(H(I,J))
         K = I
   50 CONTINUE
      EN = IGH
      T = 0.0
C     *** SEARCH FOR NEXT EIGENVALUES***
   60 IF(EN .LT. LOW) GO TO 1001
      ITS = 0
      NA = EN - 1
      ENM2 = NA - 1
C     ***LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT
C        FOR L=EN STEP -1 UNTIL LOW DO ***
   70 DO 80  LL = LOW, EN
         L = EN + LOW - LL
         IF(L .EQ. LOW) GO TO 100
         S = ABS(H(L-1,L-1)) + ABS(H(L,L))
      IF(S .EQ. 0.0) S = NORM
      IF(ABS(H(L,L-1)) .LE. MACHEP * S) GO TO 100
   80 CONTINUE
C     *** FORM SHIFT ***
  100 X = H(EN,EN)
      IF(L .EQ. EN) GO TO 270
      Y = H(NA,NA)
      W = H(EN,NA) * H(NA,EN)
      IF(L .EQ. NA) GO TO 280
      IF(ITS .EQ. 30) GO TO 1000
      IF(ITS .NE. 10 .AND. ITS .NE. 20) GO TO 130
C     *** FORM EXCEPTIONAL SHIFT ***
      T = T + X
      DO 120  I = LOW,EN
  120 H(I,I) = H(I,I) - X
      S = ABS(H(EN,NA)) + ABS(H(NA,ENM2))
      X = 0.75 * S
```

```
      Y = X                                                            HQR
      W = 0.4375 * S * S                                               HQR
  130 ITS = ITS + 1                                                    HQR
C     *** LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS***      HQR
C     *** FOR M=EN-2 STEP -1 UNTIL L DO ***                            HQR
      DO 140  MM = L, ENM2                                             HQR
         M = ENM2 + L - MM                                             HQR
         ZZ = H(M,M)                                                   HQR
         R = X - ZZ                                                    HQR
         S = Y - ZZ                                                    HQR
         P = (R * S - W) / H(M+1,M) + H(M,M+1)                         HQR
         Q = H(M+1,M+1) - ZZ - R - S                                   HQR
         R = H(M+2,M+1)                                                HQR
         S = ABS(P) + ABS(Q) + ABS(R)                                  HQR
         P = P/S                                                       HQR
         Q = Q/S                                                       HQR
         R = R/S                                                       HQR
         IF(M .EQ. L) GO TO 150                                        HQR
         IF(ABS(H(M,M-1))*(ABS(Q) + ABS(R)) .LE. MACHEP*ABS(P)         HQR
     X   * (ABS(H(M-1,M-1))+ABS(ZZ)+ABS(H(M+1,M+1)))) GO TO 150        HQR
  140 CONTINUE                                                         HQR
  150 MP2 = M + 2                                                      HQR
      DO 160  I = MP2, EN                                              HQR
         H(I,I-2) = 0.0                                                HQR
         IF(I .EQ. MP2) GO TO 160                                      HQR
         H(I,I-3) = 0.0                                                HQR
  160 CONTINUE                                                         HQR
C    * DOUBLE QR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN *     HQR
      DO 260  K = M, NA                                                HQR
         NOTLAS = K .NE. NA                                            HQR
         IF (K .EQ. M) GO TO 170                                       HQR
         P = H(K,K-1)                                                  HQR
         Q = H(K+1,K-1)                                                HQR
         R = 0.0                                                       HQR
         IF (NOTLAS) R = H(K+2,K-1)                                    HQR
         X = ABS(P) + ABS(Q) + ABS(R)                                  HQR
         IF(X .EQ. 0.0) GO TO 260                                      HQR
         P = P/X                                                       HQR
         Q = Q/X                                                       HQR
         R = R/X                                                       HQR
  170 S = SIGN(SQRT(P*P+Q*Q+R*R),P)                                    HQR
         IF(K.EQ. M) GO TO 180                                         HQR
         H(K,K-1) = -S * X                                             HQR
         GO TO 190                                                     HQR
  180    IF(L .NE. M) H(K,K-1) = -H(K,K-1)                             HQR
  190    P = P + S                                                     HQR
         X = P/S                                                       HQR
         Y = Q/S                                                       HQR
         ZZ = R/S                                                      HQR
         Q = Q/P                                                       HQR
         R = R/P                                                       HQR
C     *** ROW MODIFICATION ***                                         HQR
         DO 210  J = K, N                                              HQR
            P = H(K,J) +Q * H(K+1,J)                                   HQR
            IF(.NOT. NOTLAS) GO TO 200                                 HQR
            P = P + R * H(K+2,J)                                       HQR
            H(K+2,J) = H(K+2,J) - P * ZZ                               HQR
```

AD-A153 039 HEATING PARAMETER ESTIMATION USING COAXIAL THERMOCOUPLE 2/2
GAGES IN WIND TUN..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. N T CAHOON
UNCLASSIFIED DEC 84 AFIT/GAE/AA/84D-3 F/G 9/2 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
200       H(K+1,J) = H(K+1,J) - P * Y
          H(K,J) = H(K,J) - P * X
210    CONTINUE
       J = MIN0(EN,K+3)
C   *** COLUMN MODIFICATION ***
       DO 230  I = 1, J
          P = X * H(I,K) + Y * H(I,K+1)
          IF( .NOT. NOTLAS) GO TO 220
          P = P + ZZ * H(I,K+2)
          H(I,K+2) = H(I,K+2) - P * R
220        H(I,K+1) = H(I,K+1) - P * Q
          H(I,K) = H(I,K) - P
230    CONTINUE
C   *** ACCUMULATE TRANSFORMATIONS ***
       DO 250  I = LOW, IGH
          P = X * Z(I,K) + Y * Z(I,K+1)
          IF( .NOT. NOTLAS) GO TO 240
          P = P + ZZ * Z(I,K+2)
          Z(I,K+2) = Z(I,K+2) - P * R
240       Z(I,K+1) = Z(I,K+1) - P * Q
          Z(I,K) = Z(I,K) - P
250    CONTINUE
260 CONTINUE
    GO TO 70
C   *** ONE ROOT FOUND ***
270 H(EN,EN) = X + T
    EN = NA
    GO TO 60
C   *** TWO ROOTS FOUND ***
280 P = (Y - X)/ 2.0
    Q = P*P + W
    ZZ = SQRT(ABS(Q))
    H(EN,EN) = X + T
    X = H(EN,EN)
    H(NA,NA) = Y + T
    IF(Q .LT. 0.0) GO TO 320
C   *** REAL PAIR ***
    ZZ = P + SIGN(ZZ,P)
    X = H(EN,NA)
    S = ABS(X) + ABS(ZZ)
    P = X / S
    Q = ZZ / S
    R = SQRT(P*P + Q*Q)
    P = P / R
    Q = Q / R
C   *** ROW MODIFICATION ***
    DO 290  J = NA, N
       ZZ = H(NA,J)
       H(NA,J) = Q*ZZ + P*H(EN,J)
       H(EN,J) = Q*H(EN,J) - P*ZZ
290 CONTINUE
C   *** COLUMN MODIFICATION ***
    DO 300  I = 1, EN
       ZZ = H(I,NA)
       H(I,NA) = Q*ZZ + P*H(I,EN)
       H(I,EN) = Q*H(I,EN) - P*ZZ
300 CONTINUE
```

```
      C     *** ACCUMULATE TRANSFORMATIONS ***                          HQR    171
            DO 310  I = LOW, IGH                                        HQR    172
               ZZ = Z(I,NA)                                             HQR    173
               Z(I,NA) = Q*ZZ + P*Z(I,EN)                               HQR    174
               Z(I,EN) = Q*Z(I,EN) - P*ZZ                               HQR    175
        310 CONTINUE                                                    HQR    176
            GO TO 330                                                   HQR    177
      C     *** COMPLEX PAIR ***                                        HQR    178
        320 CONTINUE                                                    HQR    179
        330 EN = ENM2                                                   HQR    180
            GO TO 60                                                    HQR    181
      C  * SET ERROR - NO CONVERGENCE TO EIGENVALUE AFTER 30 ITERATIONS HQR    182
       1000 IERR = EN                                                   HQR    183
       1001 RETURN                                                      HQR    184
            END                                                         HQR    185
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE MULT2(N,X,Y,Z,N2)                                      MULT2     2
C    COMPUTES Z=X*Y"                                                    MULT2     3
      DIMENSION X(N2,N2),Y(N2,N2),Z(N2,N2)                              MULT2     4
      DO 20  I=1,N                                                      MULT2     5
      DO 20  J=1,N                                                      MULT2     6
      Z(I,J)= 0.                                                        MULT2     7
      DO 20  K=1,N                                                      MULT2     8
   20 Z(I,J)= Z(I,J)+X(I,K)*Y(J,K)                                      MULT2     9
      RETURN                                                            MULT2    10
      END                                                               MULT2    11
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE TRI(N,Q,X,W,Z,N2)                                      TRI    2
C    COMPUTES Z=XQX"                                                    TRI    3
      DIMENSION Q(N2,N2),X(N2,N2),W(N2,N2),Z(N2,N2)                     TRI    4
      CALL MULT(X,Q,N,W,N2)                                             TRI    5
C    X*Q IS STORED IN W                                                 TRI    6
      CALL MULT2(N,W,X,Z,N2)                                            TRI    7
C    Z=W*X"                                                             TRI    8
      RETURN                                                            TRI    9
      END                                                               TRI   10
```

```
      SUBROUTINE SGEFA(A,LDA,N,IPVT,INFO)                          SGEFA    2
      INTEGER LDA,N,IPVT(1),INFO                                   SGEFA    3
      REAL A(LDA,1)                                                SGEFA    4
C                                                                  SGEFA    5
C   SGEFA FACTORS A REAL MATRIX BY GAUSSIAN ELIMINATION.           SGEFA    6
C                                                                  SGEFA    7
C   ON ENTRY:                                                      SGEFA    8
C   A: THE MATRIX TO BE FACTORED                                   SGEFA    9
C   LDA: THE LEADING DIMENSION OF THE ARRAY A                      SGEFA   10
C   N: THE ORDER OF THE ARRAY A                                    SGEFA   11
C                                                                  SGEFA   12
C   ON RETURN                                                      SGEFA   13
C   A: AN UPPER TRIANGULAR MATRIX AND THE MULTIPLIERS              SGEFA   14
C      WHICH WERE USED TO OBTAIN IT.                               SGEFA   15
C   IPVT: AN INTEGER VECTOR OF PIVOT INDICES                       SGEFA   16
C   INFO: = 0  NORMAL VALUE.                                       SGEFA   17
C         = K  IF U(K,K).EQ. 0.0   THIS IS NOT AN ERROR            SGEFA   18
C              CONDITION FOR SGEFA, BUT INDICATES THAT             SGEFA   19
C              SGEDI WILL DIVIDE BY ZERO WHEN CALLED.              SGEFA   20
C   THIS IS FROM LINPACK USER'S GUIDE,VERSION 08/14/78             SGEFA   21
      REAL T                                                       SGEFA   22
      INTEGER ISAMAX,J,K,KP1,L,NM1                                 SGEFA   23
C                                                                  SGEFA   24
C   GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING                     SGEFA   25
      INFO= 0                                                      SGEFA   26
      NM1= N-1                                                     SGEFA   27
      IF(NM1 .LT. 1) GO TO 70                                      SGEFA   28
      DO 60  K=1,NM1                                               SGEFA   29
      KP1= K+1                                                     SGEFA   30
C                                                                  SGEFA   31
C   FIND L = PIVOT INDEX                                           SGEFA   32
      L= ISAMAX(N-K+1,A(K,K),1)+K-1                                SGEFA   33
      IPVT(K)= L                                                   SGEFA   34
C                                                                  SGEFA   35
C   ZERO PIVOT IMPLIES THIS COLUMN IS TRIANGULARIZED               SGEFA   36
      IF(A(L,K) .EQ. 0.0E0) GO TO 40                               SGEFA   37
C                                                                  SGEFA   38
C   INTERCHANGE IF NECESSARY                                       SGEFA   39
      IF(L .EQ. K) GO TO 10                                        SGEFA   40
      T=A(L,K)                                                     SGEFA   41
      A(L,K)= A(K,K)                                               SGEFA   42
      A(K,K)= T                                                    SGEFA   43
   10 CONTINUE                                                     SGEFA   44
C                                                                  SGEFA   45
C   COMPUTE MULTIPLIERS                                            SGEFA   46
      T= -1.0E0/A(K,K)                                             SGEFA   47
      CALL SSCAL(N-K,T,A(K+1,K),1)                                 SGEFA   48
C                                                                  SGEFA   49
C   ROW ELIMINATION WITH COLUMN INDEXING                           SGEFA   50
      DO 30  J=KP1,N                                               SGEFA   51
      T= A(L,J)                                                    SGEFA   52
      IF(L .EQ. K) GO TO 20                                        SGEFA   53
      A(L,J)= A(K,J)                                               SGEFA   54
      A(K,J)= T                                                    SGEFA   55
   20 CONTINUE                                                     SGEFA   56
```

```
      CALL SAXPY(N-K,T,A(K+1,K),1,A(K+1,J),1)                      SGEFA  57
   30 CONTINUE                                                      SGEFA  58
      GO TO 50                                                      SGEFA  59
   40 CONTINUE                                                      SGEFA  60
      INFO= K                                                       SGEFA  61
   50 CONTINUE                                                      SGEFA  62
   60 CONTINUE                                                      SGEFA  63
   70 CONTINUE                                                      SGEFA  64
      IPVT(N)= N                                                    SGEFA  65
      IF(A(N,N) .EQ. 0.0E0) INFO= N                                 SGEFA  66
      RETURN                                                        SGEFA  67
      END                                                           SGEFA  68
```

```
      SUBROUTINE SGEDI(A,LDA,N,IPVT,WORK)                               SGEDI   2
      INTEGER LDA,N,IPVT(1)                                             SGEDI   3
      REAL A(LDA,1),WORK(1)                                             SGEDI   4
C                                                                       SGEDI   5
C   SGEDI COMPUTES INVERSE OF MATRIX A USING                            SGEDI   6
C    FACTORS COMPUTED BY SGEFA.                                         SGEDI   7
C                                                                       SGEDI   8
C   ON ENTRY:                                                           SGEDI   9
C   A: THE OUTPUT FROM SGEFA, REAL(LDA,N)                               SGEDI  10
C   LDA: THE LEADING DIMENSION OF ARRAY A                               SGEDI  11
C   N: THE ORDER OF MATRIX A                                            SGEDI  12
C   IPVT: THE PIVOT VECTOR FROM SGEFA,INTEGER(N)                        SGEDI  13
C   WORK: WORK VECTOR,CONTENTS DESTROYED,REAL(N)                        SGEDI  14
C                                                                       SGEDI  15
C   ON RETURN:                                                          SGEDI  16
C   A: INVERSE OF THE ORIGINAL MATRIX                                   SGEDI  17
C                                                                       SGEDI  18
C   ERROR CONDITION: A DIVISION BY ZERO WILL OCCUR IF THE               SGEDI  19
C   INPUT FACTOR CONTAINS A ZERO ON THE DIAGONAL.                       SGEDI  20
C   IT WILL NOT OCCUR IF SGEFA HAS SET INFO=0                           SGEDI  21
C                                                                       SGEDI  22
C   THIS IS FROM LINPACK USER'S GUIDE,VERSION 08/14/78                  SGEDI  23
      REAL T                                                            SGEDI  24
      INTEGER I,J,K,KB,KP1,L,NM1                                        SGEDI  25
C                                                                       SGEDI  26
C   COMPUTE INVERSE                                                     SGEDI  27
      DO 100  K=1,N                                                     SGEDI  28
      A(K,K)= 1.0E0/A(K,K)                                              SGEDI  29
      T= -A(K,K)                                                        SGEDI  30
      CALL SSCAL(K-1,T,A(1,K),1)                                        SGEDI  31
      KP1= K+1                                                          SGEDI  32
      IF(N .LT. KP1) GO TO 80                                           SGEDI  33
      DO 80  J=KP1,N                                                    SGEDI  34
      T= A(K,J)                                                         SGEDI  35
      A(K,J)= 0.0E0                                                     SGEDI  36
      CALL SAXPY(K,T,A(1,K),1,A(1,J),1)                                 SGEDI  37
   80 CONTINUE                                                          SGEDI  38
   90 CONTINUE                                                          SGEDI  39
  100 CONTINUE                                                          SGEDI  40
C                                                                       SGEDI  41
C   FORM INVERSE(U)*INVERSE(L)                                          SGEDI  42
      NM1= N-1                                                          SGEDI  43
      IF(NM1 .LT. 1) GO TO 140                                          SGEDI  44
      DO 130  KB=1,NM1                                                  SGEDI  45
      K= N-KB                                                           SGEDI  46
      KP1= K+1                                                          SGEDI  47
      DO 110  I=KP1,N                                                   SGEDI  48
      WORK(I)= A(I,K)                                                   SGEDI  49
      A(I,K)= 0.0E0                                                     SGEDI  50
  110 CONTINUE                                                          SGEDI  51
      DO 120  J=KP1,N                                                   SGEDI  52
      T= WORK(J)                                                        SGEDI  53
      CALL SAXPY(N,T,A(1,J),1,A(1,K),1)                                 SGEDI  54
  120 CONTINUE                                                          SGEDI  55
      L= IPVT(K)                                                        SGEDI  56
```

```
            IF(L .NE. K) CALL SSWAP(N,A(1,K),1,A(1,L),1)           SGEDI    57
      130 CONTINUE                                                  SGEDI    58
      140 CONTINUE                                                  SGEDI    59
            RETURN                                                  SGEDI    60
            END                                                     SGEDI    61
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
      INTEGER FUNCTION ISAMAX(N,SX,INCX)                                ISAMAX     2
C   ISAMAX FINDS INDEX OF ELEMENT WITH MAX. ABSOLUTE VALUE.             ISAMAX     3
C   LINPACK USER'S GUIDE, VERSION 03/11/78                              ISAMAX     4
      REAL SX(1),SMAX                                                   ISAMAX     5
      INTEGER I,INCX,IX,N                                               ISAMAX     6
      ISAMAX= 0                                                         ISAMAX     7
      IF(N .LT. 1) RETURN                                               ISAMAX     8
      ISAMAX= 1                                                         ISAMAX     9
      IF(N .EQ. 1) RETURN                                               ISAMAX    10
      IF(INCX .EQ. 1) GO TO 20                                          ISAMAX    11
C                                                                       ISAMAX    12
C   CODE FOR INCREMENT NOT EQUAL TO 1                                   ISAMAX    13
      IX= 1                                                             ISAMAX    14
      SMAX= ABS(SX(1))                                                  ISAMAX    15
      IX= IX+INCX                                                       ISAMAX    16
      DO 10  I=2,N                                                      ISAMAX    17
      IF(ABS(SX(IX)).LE.SMAX) GO TO 5                                   ISAMAX    18
      ISAMAX= I                                                         ISAMAX    19
      SMAX= ABS(SX(IX))                                                 ISAMAX    20
    5 IX= IX+INCX                                                       ISAMAX    21
   10 CONTINUE                                                          ISAMAX    22
      RETURN                                                            ISAMAX    23
C                                                                       ISAMAX    24
C   CODE FOR INCREMENT EQUAL TO 1                                       ISAMAX    25
   20 SMAX= ABS(SX(1))                                                  ISAMAX    26
      DO 30  I=2,N                                                      ISAMAX    27
      IF(ABS(SX(I)).LE.SMAX) GO TO 30                                   ISAMAX    28
      ISAMAX= I                                                         ISAMAX    29
      SMAX= ABS(SX(I))                                                  ISAMAX    30
   30 CONTINUE                                                          ISAMAX    31
      RETURN                                                            ISAMAX    32
      END                                                               ISAMAX    33
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
      SUBROUTINE SAXPY(N,SA,SX,INCX,SY,INCY)                            SAXPY   2
C     CONSTANT TIMES A VECTOR PLUS A VECTOR.                            SAXPY   3
C     USES UNROLLED LOOP FOR INCREMENTS= 1.                             SAXPY   4
C     FROM LINPACK USER'S GUIDE,VERSION 03/11/78                        SAXPY   5
      REAL SX(1),SY(1),SA                                               SAXPY   6
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N                                 SAXPY   7
      IF(N .LE. 0) RETURN                                               SAXPY   8
      IF(SA .EQ. 0.0) RETURN                                            SAXPY   9
      IF(INCX .EQ. 1 .AND. INCY .EQ. 1) GO TO 20                        SAXPY  10
C                                                                       SAXPY  11
C     CODE FOR UNEQUAL INCREMENTS OR FOR                                SAXPY  12
C      EQUAL INCREMENTS NOT EQUAL TO 1                                  SAXPY  13
      IX= 1                                                             SAXPY  14
      IY= 1                                                             SAXPY  15
      IF(INCX.LT.0)  IX= (-N+1)*INCX +1                                 SAXPY  16
      IF(INCY.LT.0)  IY= (-N+1)*INCY +1                                 SAXPY  17
      DO 10  I=1,N                                                      SAXPY  18
      SY(IY)= SY(IY)+ SA*SX(IX)                                         SAXPY  19
      IX= IX+INCX                                                       SAXPY  20
      IY= IY+INCY                                                       SAXPY  21
   10 CONTINUE                                                          SAXPY  22
      RETURN                                                            SAXPY  23
C                                                                       SAXPY  24
C     CODE FOR BOTH INCREMENTS EQUAL TO 1                               SAXPY  25
C     CLEAN-UP LOOP                                                     SAXPY  26
   20 M=MOD(N,4)                                                        SAXPY  27
      IF(M .EQ. 0) GO TO 40                                             SAXPY  28
      DO 30  I=1,M                                                      SAXPY  29
      SY(I)= SY(I)+ SA*SX(I)                                            SAXPY  30
   30 CONTINUE                                                          SAXPY  31
      IF(N .LT. 4) RETURN                                               SAXPY  32
   40 MP1= M+1                                                          SAXPY  33
      DO 50  I=MP1,N,4                                                  SAXPY  34
      SY(I)= SY(I)+ SA*SX(I)                                            SAXPY  35
      SY(I+1)= SY(I+1)+ SA*SX(I+1)                                      SAXPY  36
      SY(I+2)= SY(I+2)+ SA*SX(I+2)                                      SAXPY  37
      SY(I+3)= SY(I+3)+ SA*SX(I+3)                                      SAXPY  38
   50 CONTINUE                                                          SAXPY  39
      RETURN                                                            SAXPY  40
      END                                                               SAXPY  41
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
      SUBROUTINE SSCAL(N,SA,SX,INCX)                                    SSCAL    2
C   SCALES A VECTOR BY A CONSTANT.                                      SSCAL    3
C   USES UNROLLED LOOPS FOR INCREMENT EQUAL TO 1.                       SSCAL    4
C   LINPACK USER'S GUIDE,VERSION 03/11/78                               SSCAL    5
C                                                                       SSCAL    6
      REAL SA,SX(1)                                                     SSCAL    7
      INTEGER I,INCX,M,MP1,N,NINCX                                      SSCAL    8
      IF(N .LE. 0) RETURN                                               SSCAL    9
      IF(INCX .EQ. 1) GO TO 20                                          SSCAL   10
C                                                                       SSCAL   11
C   CODE FOR INCREMENT NOT EQUAL TO 1                                   SSCAL   12
      NINCX= N*INCX                                                     SSCAL   13
      DO 10  I=1,NINCX,INCX                                             SSCAL   14
      SX(I)= SA*SX(I)                                                   SSCAL   15
   10 CONTINUE                                                          SSCAL   16
      RETURN                                                            SSCAL   17
C                                                                       SSCAL   18
C   CODE FOR INCREMENT EQUAL TO 1.                                      SSCAL   19
C   CLEAN-UP LOOP                                                       SSCAL   20
   20 M= MOD(N,5)                                                       SSCAL   21
      IF(M .EQ. 0) GO TO 40                                             SSCAL   22
      DO 30  I=1,M                                                      SSCAL   23
      SX(I)= SA*SX(I)                                                   SSCAL   24
   30 CONTINUE                                                          SSCAL   25
      IF(N .LT. 5) RETURN                                               SSCAL   26
   40 MP1= M+1                                                          SSCAL   27
      DO 50  I=MP1,N,5                                                  SSCAL   28
      SX(I)= SA*SX(I)                                                   SSCAL   29
      SX(I+1)= SA*SX(I+1)                                               SSCAL   30
      SX(I+2)= SA*SX(I+2)                                               SSCAL   31
      SX(I+3)= SA*SX(I+3)                                               SSCAL   32
      SX(I+4)= SA*SX(I+4)                                               SSCAL   33
   50 CONTINUE                                                          SSCAL   34
      RETURN                                                            SSCAL   35
      END                                                               SSCAL   36
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE SSWAP(N,SX,INCX,SY,INCY)                         SSWAP   2
C     INTERCHANGES TWO VECTORS.                                   SSWAP   3
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO 1.              SSWAP   4
C     LINPACK USER'S GUIDE,VERSION 03/11/78                       SSWAP   5
C                                                                 SSWAP   6
      REAL SX(1),SY(1),STEMP                                      SSWAP   7
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N                           SSWAP   8
      IF(N .LE. 0) RETURN                                         SSWAP   9
      IF(INCX .EQ. 1 .AND. INCY .EQ. 1) GO TO 20                  SSWAP  10
C                                                                 SSWAP  11
C     CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS NOT EQUAL TO 1   SSWAP  12
      IX= 1                                                       SSWAP  13
      IY= 1                                                       SSWAP  14
      IF(INCX .LT. 0)  IX= (-N+1)*INCX+1                          SSWAP  15
      IF(INCY .LT. 0)  IY= (-N+1)*INCY+1                          SSWAP  16
      DO 10  I=1,N                                                SSWAP  17
      STEMP= SX(IX)                                               SSWAP  18
      SX(IX)= SY(IY)                                              SSWAP  19
      SY(IY)= STEMP                                               SSWAP  20
      IX= IX+INCX                                                 SSWAP  21
      IY= IY+INCY                                                 SSWAP  22
   10 CONTINUE                                                    SSWAP  23
      RETURN                                                      SSWAP  24
C                                                                 SSWAP  25
C     CODE FOR BOTH INCREMENTS EQUAL TO 1.                        SSWAP  26
C     CLEAN-UP LOOP                                               SSWAP  27
   20 M= MOD(N,3)                                                 SSWAP  28
      IF(M .EQ. 0) GO TO 40                                       SSWAP  29
      DO 30  I=1,M                                                SSWAP  30
      STEMP= SX(I)                                                SSWAP  31
      SX(I)= SY(I)                                                SSWAP  32
      SY(I)= STEMP                                                SSWAP  33
   30 CONTINUE                                                    SSWAP  34
      IF(N .LT. 3) RETURN                                         SSWAP  35
   40 MP1= M+1                                                    SSWAP  36
      DO 50  I=MP1,N,3                                            SSWAP  37
      STEMP= SX(I)                                                SSWAP  38
      SX(I)= SY(I)                                                SSWAP  39
      SY(I)= STEMP                                                SSWAP  40
      STEMP= SX(I+1)                                              SSWAP  41
      SX(I+1)= SY(I+1)                                            SSWAP  42
      SY(I+1)= STEMP                                              SSWAP  43
      STEMP= SX(I+2)                                              SSWAP  44
      SX(I+2)= SY(I+2)                                            SSWAP  45
      SY(I+2)= STEMP                                              SSWAP  46
   50 CONTINUE                                                    SSWAP  47
      RETURN                                                      SSWAP  48
      END                                                         SSWAP  49
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

          SUBROUTINE FLINE(M,TSCALE,TMIN,I,ASCALE,AMIN,AYL)               FLINE      2
          DIMENSION U(12)                                                 FLINE      3
          EQUIVALENCE (T,U(1))                                            FLINE      4
          DATA IUNIT/12/                                                  FLINE      5
          REWIND IUNIT                                                    FLINE      6
          II=0                                                            FLINE      7
      100 READ(IUNIT,1000,END=190)U                                       FLINE      8
     1000 FORMAT(6E13.7)                                                  FLINE      9
          XO=(U(M)-TMIN)/TSCALE                                           FLINE     10
          YO=(U(I)-AMIN)/ASCALE+AYL                                       FLINE     11
          II=II+1                                                         FLINE     12
          IF(II.EQ.1)CALL PLOT(XO,YO,3)                                   FLINE     13
          CALL PLOT(XO,YO,2)                                              FLINE     14
          GO TO 100                                                       FLINE     15
      190 RETURN                                                          FLINE     16
          END                                                             FLINE     17
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE FLINES(M,TSCALE,TMIN,I,ASCALE,AMIN,AYL,HT,ISKIP,NCHAR)  FLINES      2
      DIMENSION U(12)                                                     FLINES      3
      EQUIVALENCE (T,U(1))                                                FLINES      4
      DATA IUNIT/12/                                                      FLINES      5
      REWIND IUNIT                                                        FLINES      6
      II=0                                                                FLINES      7
  100 READ(IUNIT,1000,END=190)U                                           FLINES      8
 1000 FORMAT(6E13.7)                                                      FLINES      9
      XO=(U(M)-TMIN)/TSCALE                                               FLINES     10
      YO=(U(I)-AMIN)/ASCALE+AYL                                           FLINES     11
      II=II+1                                                             FLINES     12
      III=((II/ISKIP))*ISKIP                                              FLINES     13
      IF(II.NE.III)GO TO 100                                              FLINES     14
      CALL SYMBOL(XO,YO,HT,NCHAR,0.,-1)                                   FLINES     15
      GO TO 100                                                           FLINES     16
  190 CONTINUE                                                            FLINES     17
      RETURN                                                              FLINES     18
      END                                                                 FLINES     19
```

```
      C  START PLOT SEQUENCE                                          FPLOT     51
            HT=.07                                                    FPLOT     52
            CALL PLOT(4.,.5,-3)                                       FPLOT     53
      C                                                               FPLOT     54
            YLAB="T2(DEG F )"                                         FPLOT     55
            TLAB="TIME(SEC) "                                         FPLOT     56
            CALL AXIS(0.,0.,TLAB,-10,TAXL,0.,TMIN,TSCALE)             FPLOT     57
            CALL AXIS(0.,0.,YLAB, 10,YAXL,90.,YMIN,YSCALE)            FPLOT     58
      C  4 IN CALL POINTS TO 4TH VARIABLE IN READ,Y                   FPLOT     59
            CALL FLINE(1,TSCALE,TMIN,4,YSCALE,YMIN,0.)                FPLOT     60
      C  5 POINTS TO Z                                                FPLOT     61
            CALL FLINES(1,TSCALE,TMIN,5,YSCALE,YMIN,0.,HT,1,3)        FPLOT     62
      C  PLOT DEPENDENT VARIABLE                                      FPLOT     63
            AYL=YAXL+1.                                               FPLOT     64
            ALAB="ALPHA(DEG)"                                         FPLOT     65
            CALL AXIS(0.,AYL,ALAB,10,AAXL,90.,AMIN,ASCALE)            FPLOT     66
      C 6 POINTS TO A                                                 FPLOT     67
            CALL FLINE(1,TSCALE,TMIN,6,ASCALE,AMIN,AYL)               FPLOT     68
      C                                                               FPLOT     69
      C                                                               FPLOT     70
      C  NEXT PLOT SEQUENCE                                           FPLOT     71
            AXO=TAXL+2.                                               FPLOT     72
            CALL PLOT(AXO,0.,-3)                                      FPLOT     73
            YLAB="   Q/QREF "                                         FPLOT     74
            YAXL=10.                                                  FPLOT     75
            YSCALE=.1                                                 FPLOT     76
            YMIN=0.                                                   FPLOT     77
            IFOX(1)=IFX(4)                                            HAROLD    56
            DO 32 I=2,6                                               HAROLD    57
             IFOX(I)=IFX(I+9)                                         HAROLD    58
        32   CONTINUE                                                 HAROLD    59
            DO 200 I=1,6                                              FPLOT     78
            IF(IFOX(I) .EQ. 0)GO TO 200                               HAROLD    60
            DATA XL/"ALPHA(DEG)","BETA(DEG) ","LOG(RE)   ",           FPLOT     80
           &"DELE(DEG) ","DELBF(DEG)","  MACH    "/                   FPLOT     81
            DATA XXL/5.,6.,4.,4.,5.,5./                               FPLOT     82
            DATA XSC/5.,1.,1.,5.,5.,5./                               FPLOT     83
            DATA XM/20.,-3.,5.,-10.,0.,0./                            FPLOT     84
            CALL AXIS(0.,0.,YLAB,10,YAXL,90.,YMIN,YSCALE)             FPLOT     85
            CALL AXIS(0.,0.,XL(I),-10,XXL(I),0.,XM(I),XSC(I))         FPLOT     86
      C  I+5 POINTS TO ALPHA   /    12 POINTS TO Q/QREF               FPLOT     87
            IPT=I+5                                                   FPLOT     88
            CALL FLINE(IPT,XSC(I),XM(I),12,YSCALE,YMIN,0.)            FPLOT     89
            CALL PLOT(AXO,0.,-3)                                      FPLOT     90
        200 CONTINUE                                                  FPLOT     91
            CALL PLOTE(N)                                             FPLOT     92
            RETURN                                                    FPLOT     93
            END                                                       FPLOT     94
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE FPLOT                                                  FPLOT      2
      COMMON/CFPLOT/IPLOT,TSCALE,TMIN,TAXL,YSCALE,YMIN,YAXL,            FPLOT      3
     $ASCALE,AMIN,AAXL                                                  FPLOT      4
      DIMENSION XL(6),XXL(6),XSC(6),XM(6)                               FPLOT      5
      DIMENSION IFOX(6)                                                 HAROLD    55
      DIMENSION DUM(1024)                                               FPLOT      6
      LOGICAL FAUTO                                                     CPARAM     2
      DIMENSION FAUTO(7)                                                OCT10      2
      DIMENSION QP(5)                                                   UPAUG16    1
      COMMON/CPARAM/HO,HALF(2),PHIC,PHIK,ZP,Z,ALPH(2),KA,S(5),          UPAUG16    2
     &CIF(5,5),KAF,IFX(5),ACC(5),IFXSUM,NPAR,DALPH(2)                   UPAUG16    3
      EQUIVALENCE (HO,QP(1))                                            UPAUG16    4
C INITIALIZE PLOTS AND WRITE PLOT FILE TO UNIT 2                        FPLOT      8
      CALL PLOTS(DUM,1024,2)                                            FPLOT      9
      CALL FACTOR(.787402)                                              FPLOT     10
      DATA IUNIT/12/                                                    FPLOT     11
      REWIND IUNIT                                                      FPLOT     12
C FIND MAX AND MINS FOR SCALING                                         FPLOT     13
      DATA TMN,TMX,YMN,YMX/1.E7,0.,5000.,-460./                         FPLOT     14
      DATA AMN,AMX/25.,45./                                             FPLOT     15
C READ T,USI1,EQUI,USI2,UMEAS,ALPHA                                     FPLOT     16
  100  READ(IUNIT,1000,END=190)T,U,V,Y,Z,A,B,R,DE,DB,DM,QN              FPLOT     17
 1000 FORMAT(6E13.7)                                                    FPLOT     18
      TMN=AMIN1(TMN,T)                                                  FPLOT     19
      TMX=AMAX1(TMX,T)                                                  FPLOT     20
      YMN=AMIN1(YMN,U,V,Y,Z)                                            FPLOT     21
      YMX=AMAX1(YMX,U,V,Y,Z)                                            FPLOT     22
      AMN=AMIN1(AMN,A)                                                  FPLOT     23
      AMX=AMAX1(AMX,A)                                                  FPLOT     24
      GO TO 100                                                         FPLOT     25
  190 CONTINUE                                                          FPLOT     26
      IF(IPLOT.GT.0)GO TO 195                                           FPLOT     27
C DEFAULT TIME AXIS LENGTH = 4 INCHES                                   FPLOT     28
      TAXL=4.                                                           FPLOT     29
      TSCALE=IFIX(((TMX-TMN)/TAXL)+.999)                                FPLOT     30
      TMIN=TMN                                                          FPLOT     31
C DEFAULT Y AXIS LENGTH =4 INCHES                                       FPLOT     32
      YAXL=4                                                            FPLOT     33
      DYMIN=25.                                                         FPLOT     34
      YSCALE=DYMIN*IFIX((YMX-YMN)/DYMIN/YAXL+1.999)                     FPLOT     35
      YMIN=YSCALE*IFIX(YMN/YSCALE)                                      FPLOT     36
C DEFAULT A AXIS LENGTH = 2 INCHES                                      FPLOT     37
      AAXL=2.                                                           FPLOT     38
      DAMIN=5                                                           FPLOT     39
      ASCALE=DAMIN*IFIX((AMX-AMN)/DAMIN/AAXL+1.999)                     FPLOT     40
      AMIN=ASCALE*IFIX(AMN/ASCALE)                                      FPLOT     41
  195 CONTINUE                                                          FPLOT     42
C                                                                       FPLOT     43
C SCALE TIME USING INPUT TSCALE ONLY                                    FPLOT     44
C PUT IN NEGATIVE OR ZERO FOR TMIN AND TAXL                             FPLOT     45
      IF(TAXL.GT.0)GO TO 198                                            FPLOT     46
      TMIN=TMN                                                          FPLOT     47
      TAXL=IFIX((TMX-TMN)/TSCALE+.999)                                  FPLOT     48
  198 CONTINUE                                                          FPLOT     49
C                                                                       FPLOT     50
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE MSCALE (N1,N2,A,X,B)                               MSCALE     2
      DIMENSION A(1), B(1)                                          MSCALE     3
      COMMON /MAIN1/ NDIM                                           MSCALE     4
      DIMENSION IRAY(6)                                             SYSTEMC    2
      DATA IRAY/6* -0/                                              SYSTEMC    3
      IRAY(4)=0                                                     SYSTEMC    4
      JEND=N2*NDIM                                                  MSCALE     5
      DO 1 I=1,N1                                                   MSCALE     6
      CALL SYSTEMC(144,IRAY)                                        OCT24      5
      DO 1 IJ=I,JEND,NDIM                                           MSCALE     7
    1 B(IJ)=X*A(IJ)                                                 MSCALE     8
      RETURN                                                        MSCALE     9
      END                                                           MSCALE    10
```

```
SUBROUTINE MMUL        74/855   OPT=0,ROUND= A/ S/ M/-D,-DS      FTN 5.1+587        84/11/19. 13.14.29         PAGE     1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE MMUL(X,Y,N1,N2,N3,Z)                                  MMUL        2
      DIMENSION X(1),Y(1),Z(1)                                         MMUL        3
      COMMON/MAIN1/NDIM                                                MMUL        4
      NEND3=NDIM*N3                                                    MMUL        5
      NEND2=NDIM*N2                                                    MMUL        6
      DO 1 I=1,N1                                                      MMUL        7
      DO 1 J=I,NEND3,NDIM                                              MMUL        8
      TM=0.                                                            MMUL        9
      K=I                                                              MMUL       10
      KK=J-I                                                           MMUL       11
    5 KK=KK+1                                                          MMUL       12
      TM=TM+X(K)*Y(KK)                                                 MMUL       13
      K=K+NDIM                                                         MMUL       14
      IF(K.LE.NEND2) GO TO 5                                           MMUL       15
    1 Z(J)=TM                                                          MMUL       16
      RETURN                                                           MMUL       17
      END                                                              MMUL       18
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
      SUBROUTINE MAT4 (N1,N2,X,Y,Z)                                MAT4     2
C     Z=YXY" X=X" IS N2XN2, Y IS N1XN2, Z IS N1XN1                 MAT4     3
      DIMENSION X(1), Y(1), Z(1)                                   MAT4     4
      COMMON /MAIN1/ NDIM                                          MAT4     5
      CALL MMUL (Y,X,N1,N2,N2,Z)                                   MAT4     6
      NN2=N2*NDIM                                                  MAT4     7
      DO 3 I=1,N1                                                  MAT4     8
      IM1=I-1                                                      MAT4     9
      II=IM1*NDIM                                                  MAT4    10
      JJ=I+II                                                      MAT4    11
      DO 2 J=I,N1                                                  MAT4    12
      TEMP=0.                                                      MAT4    13
      KK=J                                                         MAT4    14
      DO 1 K=I,NN2,NDIM                                            MAT4    15
      TEMP=TEMP+Y(K)*Z(KK)                                         MAT4    16
    1 KK=KK+NDIM                                                   MAT4    17
      Z(JJ)=TEMP                                                   MAT4    18
    2 JJ=JJ+NDIM                                                   MAT4    19
      JJ=I                                                         MAT4    20
      K=II+1                                                       MAT4    21
      KK=II+IM1                                                    MAT4    22
      DO 3 J=K,KK                                                  MAT4    23
      Z(JJ)=Z(J)                                                   MAT4    24
      JJ=JJ+NDIM                                                   MAT4    25
    3 CONTINUE                                                     MAT4    26
      RETURN                                                       MAT4    27
      END                                                          MAT4    28
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      FUNCTION XNORM (N,A)                                              XNORM       2
C         COMPUTES AN APPROXIMATION TO NORM OF A-- NOT A BOUND          XNORM       3
      DIMENSION A(1)                                                    XNORM       4
      COMMON /MAIN1/ NDIM                                               XNORM       5
      NDIM1=NDIM+1                                                      XNORM       6
      NN=N*NDIM                                                         XNORM       7
      C1=0.                                                             XNORM       8
      TR=A(1)                                                           XNORM       9
      IF (N.EQ.1) GO TO 4                                               XNORM      10
      I=2                                                               XNORM      11
      DO 2 II=NDIM1,NN,NDIM                                             XNORM      12
      J=II                                                              XNORM      13
      DO 1 JJ=I,II,NDIM                                                 XNORM      14
      C1=C1+ABS(A(J)*A(JJ))                                             XNORM      15
    1 J=J+1                                                             XNORM      16
      TR=TR+A(J)                                                        XNORM      17
    2 I=I+1                                                             XNORM      18
      TR=TR/FLOAT(N)                                                    XNORM      19
      DO 3 II=1,NN,NDIM1                                                XNORM      20
    3 C1=C1+(A(II)-TR)**2                                               XNORM      21
    4 XNORM=ABS(TR)+SQRT(C1)                                            XNORM      22
      RETURN                                                            XNORM      23
      END                                                               XNORM      24
```

```
SUBROUTINE TRANS       74/855  OPT=0,ROUND= A/ S/ M/-D,-DS     FTN 5.1+587        84/11/19. 13.14.29        PAGE      1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

                   SUBROUTINE TRANS(A,N,X)                                        TRANSP      2
                   DIMENSION A(1),X(1)                                            TRANSP      3
                   COMMON/MAIN1/NDIM                                              TRANSP      4
                   DO 10 I=1,N                                                    TRANSP      5
                   DO 10 J=1,N                                                    TRANSP      6
                   II=I+NDIM*(J-1)                                                TRANSP      7
                   JJ=J+NDIM*(I-1)                                                TRANSP      8
              10   A(II)=X(JJ)                                                    TRANSP      9
                   RETURN                                                         TRANSP     10
                   END                                                            TRANSP     11
```

```
          60  CONTINUE                                                  SGTSL  57
          70  CONTINUE                                                  SGTSL  58
          80  CONTINUE                                                  SGTSL  59
          90  CONTINUE                                                  SGTSL  60
         100  CONTINUE                                                  SGTSL  61
              RETURN                                                    SGTSL  62
              END                                                       SGTSL  63
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
      SUBROUTINE SGTSL(N,C,D,E,B,INFO)
      INTEGER N,INFO
      REAL C(1),E(1),B(1),D(1)
C
C  SGTSL GIVEN A GENERAL TRIDIAGONAL MATRIX AND A RIGHT HAND
C  SIDE WILL FIND THE SOLUTION  -  SEE THE LINPAC USER'S GUIDE
C
      INTEGER K,KB,KP1,NM1,NM2
      REAL T
      INFO=0
      C(1)=D(1)
      NM1=N-1
      IF(NM1.LT.1)GO TO 40
      D(1)=E(1)
      E(1)=0.0E0
      E(N)=0.0E0
      DO 30 K=1,NM1
      KP1=K+1
      IF(ABS(C(KP1)).LT.ABS(C(K)))GO TO 10
      T=C(KP1)
      C(KP1)=C(K)
      C(K)=T
      T=D(KP1)
      D(KP1)=D(K)
      D(K)=T
      T=E(KP1)
      E(KP1)=E(K)
      E(K)=T
      T=B(KP1)
      B(KP1)=B(K)
      B(K)=T
   10 CONTINUE
      IF(C(K).NE.0.0E0)GO TO 20
      INFO=K
      GO TO 100
   20 CONTINUE
      T=-C(KP1)/C(K)
      C(KP1)=D(KP1)+T*D(K)
      D(KP1)=E(KP1)+T*E(K)
      E(KP1)=0.0E0
      B(KP1)=B(KP1)+T*B(K)
   30 CONTINUE
   40 CONTINUE
      IF(C(N).NE.0.0E0)GO TO 50
      INFO=N
      GO TO 90
   50 CONTINUE
      NM2=N-2
      B(N)=B(N)/C(N)
      IF(N.EQ.1)GO TO 80
      B(NM1)=(B(NM1)-D(NM1)*B(N))/C(NM1)
      IF(NM2.LT.1)GO TO 70
      DO 60 KB=1,NM2
      K=NM2-KB+1
      B(K)=(B(K)-D(K)*B(K+1)-E(K)*B(K+2))/C(K)
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE INTEG(N,A,T,QUE,PHI,QD,QDT,N2)                          INTEG2      2
      DIMENSION A(N2,N2),QUE(N2,N2),PHI(N2,N2),QD(N2,N2),QDT(N2,N2)      INTEG2      3
      T2= -T*0.5                                                         INTEG2      4
      CALL MEXP(N,A(1,1),T2,PHI(1,1),QD(1,1),QDT(1,1),N2)                INTEG2      5
      CALL TRI(N,QUE(1,1),PHI(1,1),QD(1,1),QDT(1,1),N2)                  INTEG2      6
      CALL MSCALE(N,N,QDT(1,1),4.0,QDT(1,1))                             INTEG2      7
      CALL MULT(PHI(1,1),PHI(1,1),N,A(1,1),N2)                           INTEG2      8
      CALL TRI(N,QUE(1,1),A(1,1),QD(1,1),PHI(1,1),N2)                    INTEG2      9
      DO 10 I=1,N                                                        INTEG2     10
      DO 10 J=1,N                                                        INTEG2     11
   10 QDT(I,J)= QUE(I,J)+ QDT(I,J)+ PHI(I,J)                             INTEG2     12
      T6=T/6.0                                                           INTEG2     13
      CALL MSCALE(N,N,QDT(1,1),T6,QDT(1,1))                              INTEG2     14
      RETURN                                                             INTEG2     15
      END                                                                INTEG2     16
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

              SUBROUTINE HEATTUN                                                  UPOCT09    28
              COMMON/CHEAT/Q,TS,QREF,TW,M1,RENS,HBAR,HREF                         UPOCT09    29
              COMMON/COMTUN/T,TAW1,ALPHA,H,V,RHO,P,TEMP,C,TRAD,RHOG,              OCT10       3
             &TO,TSINK,XFT,DEL,PDEL                                               COMTUN      3
              LOGICAL FAUTO                                                       CPARAM      2
              DIMENSION FAUTO(7)                                                  OCT10       2
              DIMENSION QP(5)                                                     UPAUG16     1
              COMMON/CPARAM/HO,HALF(2),PHIC,PHIK,ZP,Z,ALPH(2),KA,S(5),            UPAUG16     2
             &CIF(5,5),KAF,IFX(5),ACC(5),IFXSUM,NPAR,DALPH(2)                     UPAUG16     3
              EQUIVALENCE (HO,QP(1))                                              UPAUG16     4
              COMMON/ICTPS2/TINIT(1),ERALOW,E                                     UPOCT09    32
              COMMON/CDX/DX(1)                                                    UPOCT09    33
              DATA HREF/1./                                                       UPOCT09    34
        C                                                                         UPOCT09    35
        C     CHECK FOR ALPHA SEGMENT                                             UPOCT09    36
        C                                                                         UPOCT09    37
              IF(ALPHA.LT.ALPH(2)) THEN                                           OCT10      14
1                DALPH(1)=ALPHA                                                   UPOCT09    39
1                DALPH(2)=ALPH(2)                                                 UPOCT09    40
1             ENDIF                                                               UPOCT09    41
              IF(ALPHA.GE.ALPH(2)) THEN                                           UPOCT09    42
1                DALPH(1)=ALPH(1)                                                 OCT10      15
1                DALPH(2)=ALPHA                                                   UPOCT09    43
1             ENDIF                                                               UPOCT09    44
1       C                                                                         UPOCT09    45
              HBAR=HO+HALF(1)*(DALPH(1)-ALPH(1))+HALF(2)*(DALPH(2)-ALPH(2))       UPOCT09    46
              RETURN                                                              UPOCT09    47
              END                                                                 UPOCT09    48
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

```
      SUBROUTINE TPSOSP2(DT)                                            FTPSOSP3    2
      COMMON/COSP/NPTSS,USI(6),PHI(6,6),NPT,PC(6,6),RR,                 UPAUG16     6
     &QD(6,6),QDT(6,6),QUE(6,6),A(6,6),RCX(6),RP(6),RM(6)               UPOCT09     5
      COMMON/COPT/N,IOA(8),IOB(8),IOC(8),IOFACE(8),IOD(8)               FTPSOSP3    4
      COMMON/CPC/NPTPC                                                  FTPSOSP3    5
      COMMON /MAIN1/INP,IPVT(6),WORK(6)                                 OCT10       4
      INP=6                                                             OCT10       5
      IF(DT .LE. 0.) GO TO 999                                          FTPSOSP3    7
      T=DT                                                              UPSEP24    13
      CALL INTEG(NPTPC,A,T,QUE,PHI,QD,QDT,NPTSS)                        FTPSOSP3    9
      CALL SGEFA(A(1,1),NPTSS,NPTPC,IPVT(1))                            FTPSOSP3   10
      CALL SGEDI(A(1,1),NPTSS,NPTPC,IPVT(1),WORK(1))                    FTPSOSP3   11
      CALL MAT4(NPTPC,NPTPC,QDT(1,1),A(1,1),QD(1,1))                    FTPSOSP3   12
      CALL MAT4(NPTPC,NPTPC,PC(1,1),A(1,1),QDT(1,1))                    FTPSOSP3   13
      DO 20  I=1,NPTPC                                                  FTPSOSP3   14
      DO 20  J=1,NPTPC                                                  FTPSOSP3   15
      PHI(I,J)=A(I,J)                                                   HAROLD     54
   20 PC(I,J)= QDT(I,J)+ QD(I,J)                                        FTPSOSP3   16
  999 RETURN                                                            FTPSOSP3   17
      END                                                               FTPSOSP3   18
```

```
SUBROUTINE EQUATE      74/855  OPT=0,ROUND= A/ S/ M/-D,-DS     FTN 5.1+587        84/11/19. 13.14.29        PAGE     1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

                  SUBROUTINE EQUATE (NR,NC,A,B)                                          EQUATE      2
                  DIMENSION A(1), B(1)                                                   EQUATE      3
                  COMMON /MAIN1/ NDIM                                                    EQUATE      4
                  NN=NC*NDIM                                                             EQUATE      5
                  NR1=NR-1                                                               EQUATE      6
                  DO 1 J=1,NN,NDIM                                                       EQUATE      7
                  II=J+NR1                                                               EQUATE      8
                  DO 1 IJ=J,II                                                           EQUATE      9
                  A(IJ)=B(IJ)                                                            EQUATE     10
                1 CONTINUE                                                               EQUATE     11
                  RETURN                                                                 EQUATE     12
                  END                                                                    EQUATE     13
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE GMINV (NR,NC,A,U,MR,MT)                            FGMINV    2
      DIMENSION A(1), U(1), S(30)                                   FGMINV    3
      COMMON /MAIN2/ NDIM                                           FGMINV    4
      COMMON /INOU/ KIN,KOUT                                        FGMINV    5
      NDIM1=NDIM+1                                                  FGMINV    6
      TOL=1.E-14                                                    FGMINV    7
      ADV=1.E-24                                                    FGMINV    8
      MR=NC                                                         FGMINV    9
      NRM1=NR-1                                                     FGMINV   10
      TOL1=0.                                                       FGMINV   11
      JJ=1                                                          FGMINV   12
      DO 1 J=1,NC                                                   FGMINV   13
      S(J)=DOT(NR,A(JJ),A(JJ))                                      FGMINV   14
      IF (S(J).GT.TOL1) TOL1=S(J)                                   FGMINV   15
    1 JJ=JJ+NDIM                                                    FGMINV   16
      TOL1=ADV*TOL1                                                 FGMINV   17
      ADV=TOL1                                                      FGMINV   18
      JJ=1                                                          FGMINV   19
      DO 14 J=1,NC                                                  FGMINV   20
      FAC=S(J)                                                      FGMINV   21
      JM1=J-1                                                       FGMINV   22
      JRM=JJ+NRM1                                                   FGMINV   23
      JCM=JJ+JM1                                                    FGMINV   24
      DO 2 I=JJ,JCM                                                 FGMINV   25
    2 U(I)=0.                                                       FGMINV   26
      U(JCM)=1.0                                                    FGMINV   27
      IF (J.EQ.1) GO TO 5                                           FGMINV   28
      KK=1                                                          FGMINV   29
      DO 3 K=1,JM1                                                  FGMINV   30
      IF (S(K).EQ.1.0) GO TO 3                                      FGMINV   31
      TEMP=-DOT(NR,A(JJ),A(KK))                                     FGMINV   32
      CALL VADD (K,TEMP,U(JJ),U(KK))                                FGMINV   33
    3 KK=KK+NDIM                                                    FGMINV   34
      DO 4 L=1,2                                                    FGMINV   35
      KK=1                                                          FGMINV   36
      DO 4 K=1,JM1                                                  FGMINV   37
      IF (S(K).EQ.0.) GO TO 4                                       FGMINV   38
      TEMP=-DOT(NR,A(JJ),A(KK))                                     FGMINV   39
      CALL VADD (NR,TEMP,A(JJ),A(KK))                               FGMINV   40
      CALL VADD (K,TEMP,U(JJ),U(KK))                                FGMINV   41
    4 KK=KK+NDIM                                                    FGMINV   42
      TOL1=TOL*FAC+ADV                                              FGMINV   43
      FAC=DOT(NR,A(JJ),A(JJ))                                       FGMINV   44
    5 IF (FAC.GT.TOL1) GO TO 9                                      FGMINV   45
      DO 6 I=JJ,JRM                                                 FGMINV   46
    6 A(I)=0.                                                       FGMINV   47
      S(J)=0.                                                       FGMINV   48
      KK=1                                                          FGMINV   49
      IF (S(K).EQ.0.) KK=KK+NDIM                                    FGMINV   50
      IF (S(K).EQ.0.) GO TO 8                                       FGMINV   51
      DO 7 K=1,JM1                                                  FGMINV   52
      TEMP=-DOT(K,U(KK),U(JJ))                                      FGMINV   53
      CALL VADD (NR,TEMP,A(JJ),A(KK))                               FGMINV   54
    7 KK=KK+NDIM                                                    FGMINV   55
    8 FAC=DOT(J,U(JJ),U(JJ))                                        FGMINV   56
```

```
      MR=MR-1                                                           FGMINV  57
      GO TO 11                                                          FGMINV  58
    9 S(J)=1.0                                                          FGMINV  59
      KK=1                                                              FGMINV  60
      DO 10 K=1,JM1                                                     FGMINV  61
      IF (S(K).EQ.1.) GO TO 10                                          FGMINV  62
      TEMP=-DOT(NR,A(JJ),A(KK))                                         FGMINV  63
      CALL VADD (K,TEMP,U(JJ),U(KK))                                    FGMINV  64
   10 KK=KK+NDIM                                                        FGMINV  65
   11 FAC=1./SQRT(FAC)                                                  FGMINV  66
      DO 12 I=JJ,JRM                                                    FGMINV  67
   12 A(I)=A(I)*FAC                                                     FGMINV  68
      DO 13 I=JJ,JCM                                                    FGMINV  69
   13 U(I)=U(I)*FAC                                                     FGMINV  70
   14 JJ=JJ+NDIM                                                        FGMINV  71
      IF (MR.EQ.NR.OR.MR.EQ.NC) GO TO 15                                FGMINV  72
      IF (MT.NE.0) WRITE (KOUT,19) NR,NC,MR                             FGMINV  73
   15 NEND=NC*NDIM                                                      FGMINV  74
      JJ=1                                                              FGMINV  75
      DO 18 J=1,NC                                                      FGMINV  76
      DO 16 I=1,NR                                                      FGMINV  77
      II=I-J                                                            FGMINV  78
      S(I)=0.                                                           FGMINV  79
      DO 16 KK=JJ,NEND,NDIM                                             FGMINV  80
   16 S(I)=S(I)+A(II+KK)*U(KK)                                          FGMINV  81
      II=J                                                              FGMINV  82
      DO 17 I=1,NR                                                      FGMINV  83
      U(II)=S(I)                                                        FGMINV  84
   17 II=II+NDIM                                                        FGMINV  85
   18 JJ=JJ+NDIM1                                                       FGMINV  86
      RETURN                                                            FGMINV  87
   19 FORMAT (I3,1HX,I2,8H M: RANK,I2)                                  FGMINV  88
      END                                                               FGMINV  89
```

```
   FUNCTION DOT        74/855  OPT=0,ROUND= A/ S/ M/-D,-DS      FTN 5.1+587         84/11/19. 13.14.29        PAGE     1
 DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
 FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

                      FUNCTION DOT (NR,A,B)                                                  FDOT         2
                      DIMENSION A(1), B(1)                                                   FDOT         3
                      DOT=0.                                                                 FDOT         4
                      DO 1 I=1,NR                                                            FDOT         5
                    1 DOT=DOT+A(I)*B(I)                                                      FDOT         6
                      RETURN                                                                 FDOT         7
                      END                                                                    FDOT         8
```

```
SUBROUTINE VADD        74/855  OPT=0,ROUND= A/ S/ M/-D,-DS      FTN 5.1+587        84/11/19. 13.14.29        PAGE    1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE VADD (N,C1,A,B)                                        FVADD       2
      DIMENSION A(1), B(1)                                              FVADD       3
      DO 1 I=1,N                                                        FVADD       4
    1 A(I)=A(I)+C1*B(I)                                                 FVADD       5
      RETURN                                                            FVADD       6
      END                                                               FVADD       7
```

```
SUBROUTINE ZERO        74/855  OPT=0,ROUND= A/ S/ M/-D,-DS     FTN 5.1+587        84/11/19. 13.14.29        PAGE    1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE ZERO (A,NR,NC)                                         FZERO       2
      DIMENSION A(NR,NC)                                                FZERO       3
      DO 1 IC=1,NC                                                      FZERO       4
      DO 1 IR=1,NR                                                      FZERO       5
    1 A(IR,IC)=0.0                                                      FZERO       6
      RETURN                                                            FZERO       7
      END                                                               FZERO       8
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I,ANSI=0,L=OUTS,LO=S/-A.

      SUBROUTINE VADD (N,C1,A,B)                                        FVADD       2
      DIMENSION A(1), B(1)                                              FVADD       3
      DO 1 I=1,N                                                        FVADD       4
    1 A(I)=A(I)+C1*B(I)                                                 FVADD       5
      RETURN                                                            FVADD       6
      END                                                               FVADD       7
```

BIBLIOGRAPHY

1. Neumann, R.D., Hayes, J.R.,"Advanced Techniques for the Accquisition of Aerodynamic Heating Data in Hypersonic Continuous Flow Test Facilities." AFWAL/FIMG, Wright-Patterson AFB, OH. Paper presented at 3rd Aerothermal Workshop,28 June 1984.

2. Hodge, J.K., Woo, Y.K., Cappelano, P.T.,"Parameter Estimation for Imbedded Thermocouples in Space Shuttle Wind Tunnel Test Articles with a Nonisothermal Wall." AIAA-83-1533, June 1983.

3. Knox, E.C., Martindale, W.R.,"Coax Heat Transfer Gage Usage in the VKF Tunnels B and C," VKF-TM-b.6, Sverdrup ARO, Inc. AEDC Division, January 1979.

4. Hodge, J.K.,Audley, D.R., Phillips, P.W., and Hertzler, E.K.,"Aerothermodynamic Flight Envelope Expansion for a Manned Lifting Reentry Vehicle (Space Shuttle)." Paper 3-B, AGARD CP-339, October 1982.

5. Kreith,F.,<u>Principles of Heat Transfer</u>, 2nd edition, International Textbook Company, July 1968.

6. Lutes, C.D., Hodge, J.K.,"Nonlinear Modeling and Initial Condition Estimation for Identifying the Aerothermodynamic Environment of the Space Shuttle Orbiter." AIAA-84-1749

7. Maybeck, P.S., <u>Stochastic Models, Estimation, and Control,</u> Vols I and II, Academic Press, 1979 (Vol I) and 1982 (Vol II).

8. Eykhoff,Pieter,<u>System Identification</u>, John Wiley, 1974.

9. Carslaw, H.S. and Jaeger, J.C., <u>Conduction of Heat in Solids</u>, Second Edition, Oxford, Clarendon Press, 1959.

VITA

Neil Thomas Cahoon was born on 31 July 1954 in St.Paul, Minnesota, the son of Thomas C. and Eleanor I. Cahoon. He graduated from Henry Sibley Senior High School in 1972 and the United States Air Force Academy in 1976 from which he recieved a Bachelor's Degree in Aeronautical Engineering. He completed pilot training and recieved his wings in September 1977 whereupon he was assigned as an RF4C pilot to the 67th Tactical Reconnaissance Wing, Bergstrom AFB, Texas until May 1981. He was then assigned as an RF4C instructor pilot and later as a flight examiner to the 363rd Tactical Fighter Wing, Shaw AFB, South Carolina until entering the school of Engineering, Air Force Institute of Technology, in May 1983.

permanent address: 3530 E. Marshal Gulch
Tuscon, Arizona 85718

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE AFIT/GAE/AA/84D-3	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable) AFIT/ENY	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) See Box 19	

12. PERSONAL AUTHOR(S)
Neil T. Cahoon, B.S., Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM ____ TO ____		130

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.
01	01	
20	13	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Aeronautics, Aerodynamics, Thermodynamics, Heat Transfer, Heat Transfer Coefficients, Thermal Conductivity, Stochastic Processes (~~Field 12, Group 01~~)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: Heating Parameter Estimation Using Coaxial
Thermocouple Gages in Wind Tunnel Test Articles

Thesis Advisor: James K. Hodge, Major, USAF

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER
Professional Development
(AIC)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL James K. Hodge, Major, USAF	22b. TELEPHONE NUMBER (Include Area Code) 513-255-3517 22c. OFFICE SYMBOL AFIT/ENY

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

A heat energy balance is applied to a coaxial thermocouple gage for parameter estimation in wind tunnel test articles. This method can significantly reduce wind tunnel test costs and time. Modifications to the data reduction program HEATEST (HEATing ESTimation) are made. The program allows for transient test techniques to be used as well as assuming an isothermal wall. A non-linear convective heat transfer coefficient model may also be used. Data is generated to test the new program. Temperature profiles throughout the thermocouple gage were good and were compared with changes in time step, thermocouple length, and number of discrete node points. The estimation of the convective heat transfer coefficient and thermal conductivity were excellent.

END

FILMED

5-85

DTIC